

Programmation structurée

Notion de programmation structurée-Exercices-
Correction en langage C++ /.Net

Technicien Spécialiser en Développement Informatique

2010/2011

Le développement accéléré des méthodes numériques et la disponibilité d'outils de calcul informatique de plus en plus puissants et accessibles, placent les modèles mathématiques au centre du développement scientifique technologique. Une complète maîtrise de ces méthodes et le développement d'algorithmes fiables pour la mise en œuvre s'avère donc nécessaire. Il s'agit aussi de diffuser la culture numérique auprès des scientifiques et la mettre à leur portée. Cette école s'inscrit dans ce sens.

Table des matières

| | |
|---|-----------|
| Partie 01 : Algorithme | 4 |
| 1. INTRODUCTION | 5 |
| 1.1. Notion de programme..... | 5 |
| 1.2. Le processus de la programmation | 6 |
| 2. LES VARIABLES | 7 |
| 2.1. La notion de variable | 7 |
| 2.2. Déclaration des variables | 8 |
| 2.3. Types de variables..... | 8 |
| 3. LES INSTRUCTIONS DE LECTURE ET ECRITURE | 13 |
| 4. LA STRUCTURE ALTERNATIVE | 15 |
| 4.1. Les conditions simples..... | 15 |
| 4.2. Les conditions complexes | 16 |
| 4.3. La structure alternative..... | 16 |
| 4.4. Les structures alternatives imbriquées..... | 17 |
| 4.5. Autre forme..... | 18 |
| 5. LES STRUCTURES REPETITIVES | 20 |
| 5.1. La structure POUR..... | 21 |
| 5.2. La structure TANT QUE..... | 23 |
| 5.3. La structure FAIRE | 25 |
| 6. LES TABLEAUX | 28 |
| 6.1. Les tableaux à une seul dimension | 28 |
| 6.2. Les tableaux dynamiques..... | 33 |
| 6.3. Les tableaux multidimensionnels(Matrice) | 36 |
| 7. LES STRUCTURES | 40 |
| 8. LES FONCTIONS ET PROCEDURES | 44 |
| 8.1. Les fonctions..... | 44 |
| 8.2. Les variables locales et globales..... | 47 |
| 8.3. Les passage de paramètres..... | 48 |
| 8.4. Les procédures | 48 |

| | |
|--|-----------|
| 9. <i>LES ALGORITHMES DE TRI</i> | 49 |
| 10. LES FICHIERS | 50 |
| 10.1. <i>TYPES D'ACCES</i> | 50 |
| 10.2. <i>INSTRCTIONS(fichiers texte en accès séquentiel)</i> | 50 |
| Partie 02 :Exercices | 54 |
| Partie 03 :Correction des exercices | 71 |
| 1. INTRODUCTION | 72 |
| 1.1. LES LANGAGES INFORMATIQUE | 72 |
| 1.2. <i>Correction en langage C++</i> | 73 |
| 1.3. <i>Correction en langage .NET</i> | 174 |

Partie 01 : *Algorithme*

1. INTRODUCTION

La **programmation** dans le domaine informatique est l'ensemble des activités qui permettent l'écriture des programmes informatiques. C'est une étape importante de la conception de logiciel. Pour écrire le résultat de cette activité, on utilise un langage de programmation. La programmation représente usuellement le *codage*, c'est-à-dire la rédaction du code source d'un logiciel. On utilise plutôt le terme développement pour dénoter l'ensemble des activités liées à la création d'un logiciel.

L'**algorithmique** est l'ensemble des règles et des techniques qui sont impliquées dans la définition et la conception d'algorithmes, c'est-à-dire de processus systématiques de résolution, par le calcul, d'un problème permettant de décrire les étapes vers le résultat. En d'autres termes, un algorithme est une suite finie et non-ambiguë d'opérations permettant de donner la réponse à un problème. Si les opérations d'un algorithme s'exécutent les unes après les autres, c'est un algorithme séquentiel, si elles s'exécutent en parallèle, c'est un algorithme parallèle. Si l'algorithme exploite des tâches s'exécutant sur un réseau de processeurs on parle d'algorithme réparti, ou distribué.

Le mot « algorithme » vient du nom du mathématicien Al Khawarizmi (latinisé au Moyen Âge en *Algoritmi*), qui, au IX^e siècle écrit le premier ouvrage systématique sur la solution des équations linéaires et quadratiques.

Un algorithme sert à transmettre un savoir faire. Il décrit les étapes à suivre pour réaliser un travail. Il permet d'explicitement clairement les idées de solution d'un problème indépendamment d'un langage de programmation. L'utilisateur d'un algorithme n'aura qu'à suivre toutes les instructions, dans l'ordre (en séquence) pour arriver au résultat que doit donner l'algorithme.

Le "langage algorithmique" que nous utilisons est un compromis entre un langage naturel et un langage de programmation. Nous présentons les algorithmes comme une suite d'instructions dans l'ordre des traitements. Ils sont toujours accompagnés d'un lexique qui indique, pour chaque variable, son type et son rôle. Un algorithme est délimité par les mots clés *début* et *fin*. Nous manipulerons les types couramment rencontrés dans les langages de programmation : *entier*, *réel*, *booléen*, *caractère*, *chaîne*, *tableau* et *type composite*.

Par convention, toutes les identifiants de variables seront notés en minuscule et auront un nom mnémotechnique. Il en va de même pour les fonctions, dont leur identifiant doit être le plus explicite sur son rôle. Ce dernier peut être une contraction de plusieurs mots, par conséquent pour rendre la lecture plus facile, la première lettre de chaque mot est mis en majuscule (sauf pour le premier, exemple : calculerAireRectangle).

1.1. Notion de programme

Si l'on s'intéresse aux applications de l'ordinateur, on s'aperçoit qu'elles sont très nombreuses. En

voici quelques exemples :

- Etablissement de feuille de payes, de factures
- Gestion de stocks
- Calcul de la trajectoire d'un satellite
- Suivi médical de patients dans un hôpital
- ...

Un ordinateur pour qu'il puisse effectuer des tâches aussi variées il suffit de le programmer. Effectivement l'ordinateur est capable de mettre en mémoire un programme qu'on lui fournit puis l'exécuter.

Plus précisément, l'ordinateur possède un ensemble limité d'opérations élémentaires qu'il sait exécuter. Un programme est constitué d'un ensemble de directives, nommées instructions, qui spécifient :

- ✓ les opérations élémentaires à exécuter
- ✓ la façon dont elles s'enchaînent.

Pour s'exécuter, un programme nécessite qu'on lui fournisse ce qu'on peut appelé « informations données » ou plus simplement « données ». En retour, le programme va fournir des « informations résultats » ou plus simplement résultats.

Par exemple un programme de paye nécessite des informations données : noms des employés, situations de famille, nombres d'heures supplémentaires, etc... Les résultats seront imprimés sur les différents bulletins de paye.

1.2. Le processus de la programmation

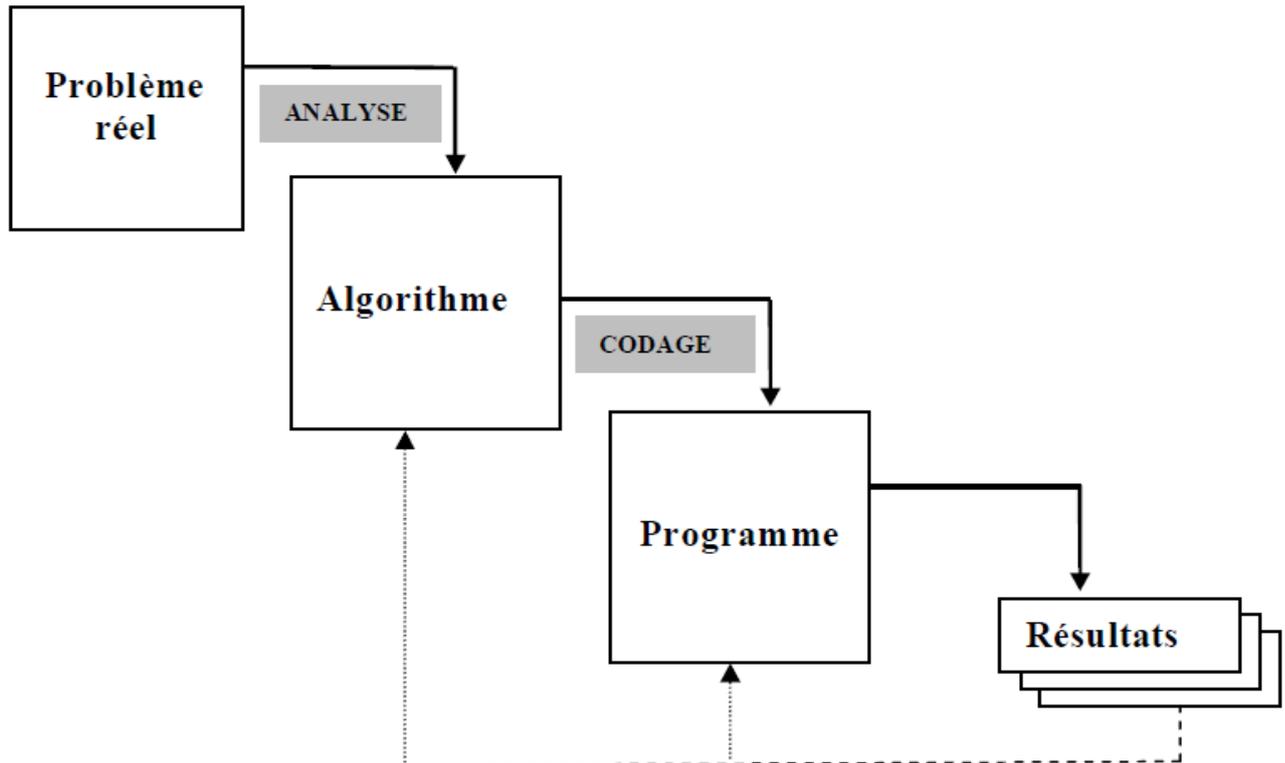
La programmation consiste, avant tout, à déterminer la démarche permettant d'obtenir, à l'aide d'un ordinateur, la solution d'un problème donné.

Le processus de la programmation se déroule en deux phases :

❖ dans un premier temps, on procède à ce qu'on appelle l'analyse du problème posé ou encore la recherche d'un algorithme¹ qui consiste à définir les différentes étapes de la résolution du problème. C'est la partie essentielle dans le processus de programmation. Elle permet de définir le contenu d'un programme en termes de données et d'actions.

❖ Dans un deuxième temps, on exprime dans un langage de programmation donné, le résultat de l'étape précédente. Ce travail, quoi qu'il soit facile, exige le respect strict de la syntaxe du langage de programmation.

Lors de l'étape d'exécution, il se peut que des erreurs syntaxiques sont signalées, ce qui entraîne des corrections en général simple ou des erreurs sémantiques plus difficiles à déceler. Dans ce dernier cas, le programme produit des résultats qui ne correspondent pas à ceux escomptés : le retour vers l'analyse sera alors inévitable.



Les différentes étapes du processus de programmation

Donc, la résolution d'un problème passe tout d'abord par la recherche d'un algorithme. L'objectif de ce cours est de vous fournir les éléments de base intervenant dans un algorithme : variable, type, instructions d'affectation, de lecture, d'écriture, structures.

2. LES VARIABLES

2.1. La notion de variable

Dans un programme informatique, on va avoir en permanence besoin de stocker provisoirement en mémoire des valeurs. Il peut s'agir de données issues du disque dur ou fournies par l'utilisateur (frappées au clavier). Il peut aussi s'agir de résultats obtenus par le programme, intermédiaires ou définitifs. Ces données peuvent être de plusieurs types (on en reparlera) : elles peuvent être des nombres, du texte, etc. Dès que l'on a besoin de stocker une information au cours d'un programme, on utilise une variable.

Une variable est un nom qui sert à repérer un emplacement donné de la mémoire, c'est à dire que la variable ce n'est qu'une adresse de mémoire.

Cette notion contribue considérablement à faciliter la réalisation des programmes. Elle permet de manipuler des données sans avoir à se préoccuper de l'emplacement qu'elles occupent effectivement en mémoire. Pour cela, il vous suffit tout simplement de leur choisir un nom. Bien entendu, la chose n'est possible que parce qu'il existe un programme de traduction (compilateur, interpréteur) de votre programme dans le langage machine ; c'est lui qui attribuera une adresse à chaque variable.

Le programmeur ne connaît que les noms A, MONTANT, RACINE... Il ne se préoccupe pas des adresses qui leur sont attribuées en mémoires.

Le nom (on dit aussi identificateur) d'une variable, dans tous les langages, est formé d'une ou plusieurs lettres ; les chiffres sont également autorisés à condition de ne pas apparaître au début du nom. La plupart des signes de ponctuation sont exclus en particulier les espaces.

Par contre, le nombre maximum de caractères autorisés varie avec les langages. Il va de deux dans certains langages jusqu'à quarante.

Dans ce cours, aucune contrainte de longueur ne vous sera imposée. De même nous admettrons que les lettres peuvent être indifférentes des majuscules ou des minuscules.

Remarque : Pour les noms des variables choisissez des noms représentatifs des informations qu'ils désignent ; ainsi MONTANT est un meilleur choix que X pour désigner le montant d'une facture.

Une variable peut être caractérisé aussi par sa valeur. A un instant donné, une variable ne peut contenir qu'une seule valeur. Bien sûr, cette valeur pourra évoluer sous l'action de certaines instructions du programme.

Outre le nom et la valeur, une variable peut être caractérisée par son type. Le type d'une variable définit la nature des informations qui seront représentées dans les variables (numériques, caractères...).

Ce type implique des limitations concernant les valeurs qui peuvent être représentées. Il limite aussi les opérations réalisables avec les variables correspondantes. Ainsi, les opérations arithmétiques (addition, soustraction, multiplication, division) possibles des variables numériques, n'ont aucun sens pour des variables de type caractères. Par contre les comparaisons seront possibles pour les deux types.

2.2. Déclaration des variables

La première chose à faire tout au début de l'algorithme, avant de pouvoir utiliser des variables, c'est de faire la déclaration des variables.

Lorsqu'on déclare une variable, on lui attribue un nom et on lui réserve un emplacement mémoire. La taille de cet emplacement mémoire dépend du type de variable. C'est pour cette raison qu'on doit préciser lors de la déclaration le type du variable.

La syntaxe d'une déclaration de variable est la suivante :

VARIABLE nom : TYPE

ou

VARIABLES nom1, nom2, ... : TYPE

2.3. Types de variables

2.3.1. Type numérique

Commençons par le cas très fréquent, celui d'une variable destinée à recevoir des nombres.

Généralement, les langages de programmation offrent les types suivants :

✦ *ENTIER*

Le type entier désigne l'ensemble des nombres entiers négatifs ou positifs dont les valeurs varient entre -32 768 à 32 767.

On écrit alors :

VARIABLES i, j, k : ENTIER

✦ *REEL*

Le type réel comprend les variables numériques qui ont des valeurs réelles. La plage des valeurs du type réel est :

-3,40x10³⁸ à -1,40x10⁴⁵ pour les valeurs négatives

1,40x10⁻⁴⁵ à 3,40x10³⁸ pour les valeurs positives

On écrit alors :

VARIABLES x, y : REEL

Remarque : Le type de variable choisi pour un nombre va déterminer les valeurs maximales et minimales des nombres pouvant être stockés dans la variable. Elle détermine aussi la précision de ces nombres (dans le cas de nombres décimaux).

2.3.2. Type chaîne

En plus, du type numérique on dispose également du type chaîne (également appelé caractère ou alphanumérique).

Dans une variable de ce type, on stocke des caractères, qu'il s'agisse de lettres, de signes de ponctuation, d'espaces, ou même de chiffres. Le nombre maximal de caractères pouvant être stockés dans une seule variable chaîne dépend du langage utilisé.

On écrit alors :

VARIABLE nom, prenom, adresse : CHAINE

Une chaîne de caractères est notée toujours soit entre guillemets, soit entre des apostrophes.

Cette notation permet d'éviter les confusions suivantes :

× Confondre un chiffre et une suite de chiffres. Par exemple, 423 peut représenter le nombre 423 (quatre cent vingt-trois), ou la suite de caractères 4, 2, et 3.

× La confusion qui consiste à ne pas pouvoir faire la distinction entre le nom d'une variable et son contenu.

Remarque : Pour les valeurs des variables de type chaîne, il faut respecter la casse. Par exemple, la

Chaîne " Salut " est différente de la chaîne "salut ".

2.3.3. Type booléen

Dans ce type de variables on y stocke uniquement des valeurs logiques VRAI ou FAUX, TRUE ou FALSE, 0 ou 1.

On écrit alors :

VARIABLE etat : BOOLEEN

2.3.4. Opérateurs et expressions

2.3.4.1. Opérateurs

Un opérateur est un signe qui relie deux variables pour produire un résultat.

Les opérateurs dépendent des types de variables mis en jeu.

Pour le type numérique on a les opérateurs suivants :

+ : Addition

- : Soustraction

* : Multiplication

/ : Division

^ : Puissance

Tandis que pour le type chaîne, on a un seul opérateur qui permet de concaténer deux chaînes de caractères. Cet opérateur de concaténation est noté &.

Par exemple : la chaîne de caractères " Salut " concaténer à la chaîne "tout le monde" donne comme résultat la chaîne "Salut tout le monde".

2.3.4.2. Expressions

Une expression est un ensemble de variables (ou valeurs) reliées par des opérateurs et dont la Valeur du résultat de cette combinaison est unique.

Par exemple :

$7 \quad 5+4 \quad x + 15 - y/2 \quad \text{nom} \ \& \ \text{prenom}$

où x et y sont des variables numériques (réels ou entiers) et nom et prenom sont des variables chaîne.

Dans une expression où on y trouve des variables ou valeurs numériques, l'ordre de priorité des opérateurs est important. En effet, la multiplication et la division sont prioritaires par rapport à l'addition et la soustraction.

Par exemple, $12 * 3 + 5$ donne comme résultat 41.

Si l'on veut modifier cette ordre de priorité on sera obligé d'utiliser les parenthèse.

Par exemple, $12 * (3 + 5)$ donne comme résultat 96.

2.3.5. L'instruction d'affectation

L'instruction d'affectation est opération qui consiste à attribuer une valeur à une variable. On la notera avec le signe \leftarrow

Cette instruction s'écrit :

$VARIABLE \leftarrow valeur$

Par exemple :

$MONTANT \leftarrow 3500.$

On dit qu'on affecte (ou on attribue) la valeur 3500 à la variable numérique MONTANT.

Si dans une instruction d'affectation, la variable à laquelle on affecte la valeur et la valeur affectée ont des types différents, cela provoquera une erreur.

On peut aussi attribuer à une variable la valeur d'une variable ou d'une expression de façon générale.

On écrit :

$VARIABLE \leftarrow EXPRESSION$

Par exemple :

$A \leftarrow B$

$A \leftarrow B * 2 + 5$

Dans ce cas, l'instruction d'affectation sera exécutée en deux temps :

- D'abord, on calcule la valeur de l'expression

- On affecte la valeur obtenue à la variable à gauche.

On peut même avoir des cas où la variable de gauche qui figure dans l'expression à droite.

Par exemple :

$A \leftarrow A + 5$

Dans cette exemple, après l'exécution de l'instruction d'affectation la valeur de la variable A sera augmenter de 5.

Remarque :

Dans une instruction d'affectation on a toujours :

- à gauche de la flèche d'affectation un nom de variable
- à droite de la flèche d'affectation une valeur ou une expression
- l'expression à droite de la flèche doit être du même type que la variable située à gauche.

Si dans une instruction d'affectation une ces points n'est pas respecté, cela engendra une erreur.

Il est à noter que l'ordre dans lequel sont écrites les instructions est essentiel dans le résultat final.

Exemple :

| CAS I | CAS II |
|-------------------|-------------------|
| $A \leftarrow 15$ | $A \leftarrow 30$ |
| $A \leftarrow 30$ | $A \leftarrow 15$ |

Après exécution des deux instructions d'affectation, la valeur de A sera :

- Cas I : 30
- Cas II : 15

Exercices

1. Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

Entier : A, B

$A \leftarrow 1$

$B \leftarrow A + 3$

$A \leftarrow 3$

2. Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?

Entier : A, B, C

$A \leftarrow 5$

$B \leftarrow 3$

$C \leftarrow A + B$

$A \leftarrow 2$

$C \leftarrow B - A$

3. Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

Entier : A, B

$A \leftarrow 5$

$B \leftarrow A + 4$

$A \leftarrow A + 1$

$B \leftarrow A - 4$

4. Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?

Entier : A, B, C

$A \leftarrow 3$

$B \leftarrow 10$

$C \leftarrow A + B$

$B \leftarrow A + B$

$A \leftarrow C$

5. Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

Entier : A, B

$A \leftarrow 5$

$B \leftarrow 2$

$A \leftarrow B$

$B \leftarrow A$

Questions : les deux dernières instructions permettent-elles d'échanger les deux valeurs de B et A ? Si l'on inverse les deux dernières instructions, cela change-t-il quelque chose ?

6. Ecrire un algorithme permettant d'échanger les valeurs de deux variables A et B, et ce quel que soit leur contenu préalable.

7. On dispose de trois variables A, B et C. Ecrivez un algorithme transférant à B la valeur de A, à C la valeur de B et à A la valeur de C (toujours quels que soient les contenus préalables de ces variables).

8. Que produit l'algorithme suivant ?

Caractères : A, B, C

$A \leftarrow \text{"423"}$

B ← "12"
C ← A + B

9. Que produit l'algorithme suivant ?

Caractères : A, B

A ← "423"
B ← "12"
C ← A & B

Solutions

1.

| Après exécution de l'instruction | La valeur des variables est : |
|----------------------------------|-------------------------------|
| A ← 1 | A = 1 B = ? |
| B ← A + 3 | A = 1 B = 4 |
| A ← 3 | A = 3 B = 4 |

2.

| Après exécution de l'instruction | La valeur des variables est : |
|----------------------------------|-------------------------------|
| A ← 5 | A = 5 B = ? C = ? |
| B ← 3 | A = 5 B = 3 C = ? |
| C ← A + B | A = 5 B = 3 C = 8 |
| A ← 2 | A = 2 B = 3 C = 8 |
| C ← B - A | A = 2 B = 3 C = 1 |

3.

| Après exécution de l'instruction | La valeur des variables est : |
|----------------------------------|-------------------------------|
| A ← 5 | A = 5 B = ? |
| B ← A + 4 | A = 5 B = 9 |
| A ← A + 1 | A = 6 B = 9 |
| B ← A - 4 | A = 6 B = 2 |

4.

| Après exécution de l'instruction | La valeur des variables est : |
|----------------------------------|-------------------------------|
| A ← 3 | A = 3 B = ? C = ? |
| B ← 10 | A = 3 B = 10 C = ? |
| C ← A + B | A = 3 B = 10 C = 13 |
| B ← A + B | A = 3 B = 13 C = 13 |
| A ← C | A = 13 B = 13 C = 13 |

5.

| Après exécution de l'instruction | La valeur des variables est : |
|----------------------------------|-------------------------------|
| A ← 5 | A = 5 B = ? |

Programmation structurée

| | |
|------------------|-----------------|
| $B \leftarrow 2$ | $A = 5 \ B = 2$ |
| $A \leftarrow B$ | $A = 2 \ B = 2$ |
| $B \leftarrow A$ | $A = 2 \ B = 2$ |

Les deux dernières instructions ne permettent donc pas d'échanger les deux valeurs de B et A, puisque l'une des deux valeurs (celle de A) est ici écrasée.

Si l'on inverse les deux dernières instructions, cela ne changera rien du tout, hormis le fait que cette fois c'est la valeur de B qui sera écrasée.

6. L'algorithme est :

$C \leftarrow A$
 $A \leftarrow B$
 $B \leftarrow C$

On est obligé de passer par une variable dite temporaire (la variable C).

7. L'algorithme est :

$D \leftarrow C$
 $C \leftarrow B$
 $B \leftarrow A$
 $A \leftarrow D$

En fait, quel que soit le nombre de variables, une seule variable temporaire suffit.

8. Il ne peut produire qu'une erreur d'exécution, puisqu'on ne peut pas additionner des caractères.

9. On peut concaténer ces variables. A la fin de l'algorithme, C vaudra donc "42312".

3. LES INSTRUCTIONS DE LECTURE ET ECRITURE

Considérons le programme suivant :

ENTIER : A
 $A \leftarrow 12 \wedge 2$

Il permet de calculer le carré de 12.

Le problème de ce programme, c'est que, si l'on veut calculer le carré d'un autre nombre que 12, il faut réécrire le programme.

D'autre part, la machine calcule le résultat et l'utilisateur qui fait exécuter ce programme ne saura jamais que ce résultat correspond au carré de 12.

C'est pour cela qu'il faut introduire des instructions qui permettent le dialogue avec l'utilisateur.

En effet, il existe une instruction qui permet à l'utilisateur de faire entrer des valeurs au clavier pour qu'elles soient utilisées par le programme. La syntaxe de cette instruction de lecture est :

LIRE NomVariable

Lorsque le programme rencontre une instruction LIRE, l'exécution du programme s'interrompt, attendant la saisie d'une valeur au clavier.

Dès que l'on frappe sur la touche ENTER, l'exécution reprend.

Une autre instruction permet au programme de communiquer des valeurs à l'utilisateur en les affichant à l'écran. La syntaxe de cette instruction d'écriture est :

AFFICHER NomVariable OU **ECRIRE** NomVariable

ou de façon générale

AFFICHER Expression OU **ECRIRE** Expression

Remarque : Avant de lire une variable, il est fortement conseillé d'écrire des libellés à l'écran, afin de prévenir l'utilisateur de ce qu'il doit frapper. La même chose pour l'instruction d'écriture.

Exemple :

```
Réels : A, CARRE
AFFICHER "Entrez un nombre"
Lire A
CARRE ← A * A
AFFICHER "Le carré de ce nombre est : "
AFFICHER CARRE
```

Exercices

1. Quel résultat produit le programme suivant ?

```
ENTIERS: Val, Double
Val ← 231
Double ← Val * 2
AFFICHER Val
AFFICHER Double
```

2. Ecrire un programme qui demande deux nombres entiers à l'utilisateur, puis qui calcule et affiche le somme de ces nombres.

3. Ecrire un programme qui lit le prix HT d'un article, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC correspondant. Faire en sorte que des libellés apparaissent clairement.

4. Ecrire un programme qui lit une valeur et qui nous calcule l'inverse de cette valeur.

5. Le surveillant général d'un établissement scolaire souhaite qu'on lui écrit un programme qui calcule, pour chaque élève, la moyenne des notes des cinq matières. Ces matières sont avec leur coefficient :

| MATIERE | COEFFICIENT |
|-----------------------|-------------|
| Math | 5 |
| Physique | 5 |
| Français | 4 |
| Anglais | 2 |
| Histoire - Géographie | 2 |

Corrections

1. On verra apparaître à l'écran :

| |
|------------|
| 231 462 |
|------------|

2. Le programme est :

```
ENTIERS : A, B, SOMME
AFFICHER "Entrez le premier nombre"
Lire A
AFFICHER " Entrez le deuxième nombre"
Lire B
SOMME ← A + B
AFFICHER "La somme de ces deux nombres est : "
```

AFFICHER SOMME

Remarque : On peut remplacer les deux derniers lignes par :

AFFICHER "La somme de ces deux nombres est : " & SOMME

OU

AFFICHER "La somme de ces deux nombres est : " , SOMME

3. Le programme est :

REELS : pht, ttva, pttc

ENTIER : nb

AFFICHER "Entrez le prix hors taxes :"

LIRE pht

AFFICHER "Entrez le nombre d'articles :"

LIRE nb

AFFICHER "Entrez le taux de TVA :"

LIRE ttva

$Pttc \leftarrow nb * pht * (1 + ttva)$

AFFICHER "Le prix toutes taxes est : ", ttva

4. Le programme est :

REELS : x, inverse

AFFICHER "Entrez une valeur :"

LIRE x

$inverse \leftarrow 1 / x$

AFFICHER "L'inverse est : ", inverse

5. Le programme est :

REELS : mat, phy, ang, fra, hg, moyenne

AFFICHER "Entrez la note de math :"

LIRE mat

AFFICHER "Entrez la note de physique :"

LIRE phy

AFFICHER "Entrez la note de français :"

LIRE fra

AFFICHER "Entrez la note d'anglais :"

LIRE ang

AFFICHER "Entrez la note d'histoire-Géo :"

LIRE hg

$moyenne \leftarrow ((mat + phy) * 5 + fra * 4 + (ang + hg) * 2) / 18$

AFFICHER "La moyenne est : ", moyenne

4. LA STRUCTURE ALTERNATIVE

4.1. Les conditions simples

Une condition simple consiste en une comparaison entre deux expressions du même type.

Cette comparaison s'effectue avec des opérateurs de comparaison. Voici la liste de ces opérateurs

accompagnés de leur signification dans le cas des types numérique ou chaîne :

| Opérateur | Signification numérique | Signification chaîne |
|-----------|-------------------------|----------------------|
| = | égal à | égal à |

Programmation structurée

| | | |
|----|-------------------|---|
| <> | différent | différent |
| < | inférieur | placé avant dans l'ordre alphabétique |
| > | supérieur | placé après dans l'ordre alphabétique |
| <= | inférieur ou égal | placé avant dans l'ordre alphabétique ou égal |
| >= | supérieur ou égal | placé après dans l'ordre alphabétique ou égal |

Pour la comparaison du type chaîne c'est l'ordre alphabétique qu'est utilisé dans le cas où l'on compare deux lettres majuscules ou minuscules. Mais si l'on compare majuscules et minuscules, il faut savoir que les majuscules apparaissent avant les minuscules. Ainsi, par exemple : "M" < "m".

4.2. Les conditions complexes

Certains problèmes exigent parfois de formuler des conditions qui ne peuvent pas être exprimées sous la forme simple vu en dessus. A cet effet, la plupart des langages autorisent des conditions formées de plusieurs conditions simples reliées entre elles par ce qu'on appelle des opérateurs logiques. Ces opérateurs sont : **ET**, **OU** et **NON**.

- Pour que la condition complexe,

condition1 **ET** *condition2*

soit VRAI, il faut impérativement que la *condition1* soit VRAI et que la *condition2* soit VRAI.

- Pour que la condition

condition1 **OU** *condition2*

soit VRAI, il suffit que *condition1* soit VRAI ou *condition2* soit VRAI. Il est à noter que cette condition complexe sera VRAI si *condition1* et *condition2* sont VRAI.

- Le NON inverse une condition :

NON(*condition*)

est VRAI si condition est FAUX, et il sera FAUX si condition est VRAI.

D'une manière générale, les opérateurs logiques peuvent porter, non seulement sur des conditions simples, mais aussi sur des conditions complexes. L'usage de parenthèses permet dans de tels cas de régler d'éventuels problèmes de priorité. Par exemple, la condition :

(a < 0 **ET** b > 1) **OU** (a > 0 **ET** b > 3)

est VRAI si l'une au moins des conditions entre parenthèses est VRAI.

4.3. La structure alternative

Supposons que nous avons besoin, dans un programme, d'afficher un message précisant que la valeur d'une variable est positive ou négative. Avec les instructions de base que nous avons vu (celles qui permettent la manipulation des variables : affectation, lecture, écriture), on ne peut pas. Il faut introduire une des instructions de structuration du programme (ces instructions servent à préciser comment doivent s'enchaîner chronologiquement ces instructions de base) qui donne la possibilité d'effectuer des choix dans le traitement réalisé. Cette instruction s'appelle la structure alternative. Sa syntaxe est :

SI *condition* **ALORS**
bloc 1 d'instructions
SINON
bloc 2 d'instructions
FIN SI

Si la condition mentionnée après **SI** est VRAI, on exécute le *bloc1 d'instructions* (ce qui figure après le mot **ALORS**); si la condition est fausse, on exécute le *bloc2 d'instructions* (ce qui figure après le mot **SINON**).

Exemple :

```
SI a > 0 ALORS  
AFFICHER "valeur positive"  
SINON  
AFFICHER "valeur négative"  
FIN SI
```

Dans ce programme, on vérifie si la valeur de a est supérieure à 0, on affichera le message "valeur positive". Dans le cas contraire, il sera affiche le message "valeur négative".

La structure alternative peut prendre une autre forme possible où l'une des parties du choix est absente. Elle s'écrit dans ce cas :

```
SI condition ALORS  
    bloc d'instructions  
FIN SI
```

Exemple : Dans un programme de calcul du montant d'une facture, on applique une remise de 1% si le montant dépasse 5000 Dhs. Nous écrivons :

```
SI montant > 5000 ALORS  
    montant ← montant * 0.99  
FIN SI
```

4.4. Les structures alternatives imbriquées

Il peut arriver que l'une des parties d'une structure alternative contienne à son tour une structure alternative. Dans ce cas, on dit qu'on a des structures alternatives imbriquées les unes dans les autres.

Exemple : Ecrire un programme qui donne l'état de l'eau selon sa température.

```
Entier : Temp  
AFFICHER "Entrez la température de l'eau :"  
Lire Temp  
Si Temp =< 0 Alors  
    AFFICHER "C'est de la glace"  
Sinon  
    Si Temp < 100 Alors  
        AFFICHER "C'est du liquide"  
    Sinon  
        AFFICHER "C'est de la vapeur"  
Finsi  
Finsi
```

On peut aussi écrire :

```
Entier : Temp  
AFFICHER "Entrez la température de l'eau :"  
Lire Temp  
Si Temp =< 0 Alors  
    AFFICHER "C'est de la glace"  
Finsi
```

```
Temp > 0 Et Temp < 100 Alors  
AFFICHER "C'est du liquide"  
Finsi  
Si Temp > 100 Alors  
AFFICHER "C'est de la vapeur"  
Finsi
```

La première version est plus simple à écrire et plus lisible. Elle est également plus performante à l'exécution. En effet, les conditions se ressemblent plus ou moins, et surtout on oblige la machine à examiner trois tests successifs alors que tous portent sur la même chose, la valeur de la variable Temp.

Mais aussi, et surtout, nous avons fait des économies sur le temps d'exécution de l'ordinateur. Si la température est inférieure à zéro, celui-ci écrit « C'est de la glace » et passe directement à la fin, sans être ralenti par l'examen des autres possibilités.

4.5. Autre forme

Dans des langages de programmation, la structure alternative peut prendre une autre forme qui permet d'imbriquer plusieurs. Sa syntaxe est :

```
SELON expression  
  valeur1 : action1  
  valeur2 : action2  
  ...  
  valeurN : actionN  
SINON : action  
FIN SELON
```

Si *expression* est égale à *valeuri*, on exécute action et on passe à la suite de l'algorithme. Sinon on exécute action et on passe à la suite de l'algorithme.

Exercices

1. Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif (on laisse de côté le cas où le produit est nul). Attention toutefois : on ne doit pas calculer le produit des deux nombres.
2. Ecrire un algorithme qui demande trois noms à l'utilisateur et l'informe ensuite s'ils sont rangés ou non dans l'ordre alphabétique.
3. Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on inclut cette fois le traitement du cas où le nombre vaut zéro).
4. Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif ou positif (on inclut cette fois le traitement du cas où le produit peut être nul). Attention toutefois, on ne doit pas calculer le produit !
5. Ecrire un algorithme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :
 - « Poussin » de 6 à 7 ans
 - « Pupille » de 8 à 9 ans
 - « Minime » de 10 à 11 ans
 - « Cadet » après 12 ans
6. a partir d'un montant lu, on détermine un montant net par application d'une remise de :
 - 1% si le montant est compris entre 2000 et 5000 Dhs (valeurs comprises)
 - 2 % si le montant est supérieur à 5000 Dhs.

Solutions

1. Le programme est :

```
Entier : m, n
AFFICHER "Entrez deux nombres : "
Lire m, n
Si  $m * n > 0$  Alors
    AFFICHER "Leur produit est positif"
Sinon
    Ecrire "Leur produit est négatif"
Finsi
```

2. Le programme est :

```
Caractère : a, b, c
Afficher "Entrez successivement trois noms : "
Lire a, b, c
Si  $a < b$  et  $b < c$  Alors
    Afficher "Ces noms sont classés alphabétiquement"
Sinon
    Afficher "Ces noms ne sont pas classés"
Finsi
```

3. Le programme est :

```
Entier : n
Afficher "Entrez un nombre : "
Lire n
Si  $n < 0$  Alors
    Afficher "Ce nombre est négatif"
Sinon
    Si  $n = 0$  Alors
        Afficher "Ce nombre est nul"
    Sinon
        Afficher "Ce nombre est positif"
    Finsi
Finsi
```

4. Le programme est :

```
Entier : m, n
Afficher "Entrez deux nombres : "
Lire m, n
Si  $m = 0$  OU  $n = 0$  Alors
    Afficher "Le produit est nul"
Sinon
    Si  $(m < 0$  ET  $n < 0)$  OU  $(m > 0$  ET  $n > 0)$  Alors
        Afficher "Le produit est positif"
    Sinon
        Afficher "Le produit est négatif"
    Finsi
Finsi
```

5. Le programme est :

```
Entier : age:
Afficher "Entrez l'âge de l'enfant : "
Lire age
Si age >= 12 Alors
    Afficher "Catégorie Cadet"
Sinon
    Si age >= 10 Alors
        Afficher "Catégorie Minimale"
    Sinon
        Si age >= 8 Alors
            Afficher "Catégorie Pupille"
        Sinon
            Si age >= 6 Alors
                Afficher "Catégorie Poussin"
            Finsi
        Finsi
    Finsi
Finsi
```

6. Le programme est :

```
Réels : montant , taux , remise
Afficher "Entrez le montant : "
Lire montant
Si montant < 2000 Alors
    taux ← 0
Sinon
    Si montant <= 5000 Alors
        taux ← 1
    Sinon
        taux ← 2
    Fin si
Fin si
```

```
Montant ← montant * (1 - taux / 100)
Afficher "Le montant net est : ", montant
```

5. LES STRUCTURES REPETITIVES

Reprenons le programme du surveillant général qui calcule la moyenne des notes. L'exécution de ce programme fournit la moyenne des notes uniquement pour un seul élève. S'il l'on veut les moyennes de 200 élèves, il faut ré exécuter ce programme 200 fois. Afin d'éviter cette tâche fastidieux d'avoir ré exécuter le programme 200 fois, on peut faire recourt à ce qu'on appelle les **structures répétitives**.

On dit aussi les **structures itératives** ou **boucles**.

Une structure répétitive sert à répéter un ensemble d'instructions. Il existe trois formes de structures répétitives : **POUR, TANT QUE, FAIR**.

5.1. La structure POUR

Cette structure permet de répéter des instructions un nombre connu de fois. Sa syntaxe est :

```
POUR compteur = valeur initial A valeur final PAS DE incrément  
    Instructions à répéter  
FIN POUR
```

compteur c'est ce qu'on appelle **compteur**. C'est une variable de type entier.

Valeur *initial* et *valeur final* sont respectivement les valeur initiale et final prise par le compteur. Ce sont des valeurs entières.

incrément est la valeur d'augmentation progressive du compteur. La valeur par défaut du pas est de 1.

Dans de telle on peut ne pas le préciser.

Remarques :

Pour un pas positif, la valeur négative doit être inférieure à la valeur finale. Pour un pas négatif, valeur négative doit être supérieure à la valeur finale.

Si la valeur initiale est égale à la valeur finale, la boucle sera exécutée une seule fois.

Exemple : Réécrivons le programme du surveillant général de façon qu'il puisse calculer les moyennes de 200 élèves.

```
    REELS : mat, phy, ang, fra, hg, moyenne  
    ENTIER : i:  
POUR i = 1 A 200 (PAS 1)  
    Afficher "Entrez la note de math :"  
    LIRE mat  
    Afficher "Entrez la note de physique :"  
    LIRE phy  
    Afficher "Entrez la note de français :"  
    LIRE fra  
    Afficher "Entrez la note 'anglais :"  
    LIRE ang  
    Afficher "Entrez la note d'histoire-Géo :"  
    LIRE hg  
    moyenne ← ((mat + phy) * 5 + fra * 4 + (ang+ hg) * 2) / 18  
    Afficher "La moyenne est : ", moyenne  
FIN POUR
```

Exercices

1. Ecrire un algorithme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 7) :

Table de 7 :

$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

$$7 \times 3 = 21$$

...

$$7 \times 10 = 70$$

2. Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer :

$$1 + 2 + 3 + 4 + 5 = 15$$

3. Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle.

NB : la factorielle de 8, notée $8!$ vaut $1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$

4. Ecrire un algorithme qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres :

Entrez le nombre numéro 1 : 12

Entrez le nombre numéro 2 : 14

...

Entrez le nombre numéro 20 : 6

Le plus grand de ces nombres est : 14

Modifiez ensuite l'algorithme pour que le programme affiche de surcroît en quelle position avait été saisie ce nombre :

C'était le nombre numéro 2

5. Ecrire un algorithme qui :

- lit d'abord une valeur

- ensuite il va lire successivement 20 nombres.

- enfin il va déterminer combien de fois la première valeur a été saisie (sans compter la première saisie).

Solutions

1. Le programme est :

Entiers : i , valeur

Lire valeur

POUR i = 1 **A** valeur

Afficher valeur & " X " & i & " = " & valeur * i

FIN POUR

2. Le programme est :

Entiers : i , valeur , somme

Lire valeur

somme \leftarrow 0

POUR i = 1 **A** valeur

 somme \leftarrow somme + i

FIN POUR

Afficher "La somme des " & valeur & " premiers entiers est : " & somme

3. Le programme est :

Entiers : i , valeur , factoriel

Lire valeur

factoriel \leftarrow 1

POUR i = 1 **A** valeur

 factoriel \leftarrow factoriel * i

FIN POUR

Afficher "Le factoriel de " & valeur & " est : " & factoriel

4. Le programme est :

Entiers i , a , max , pmax

Afficher " Entrez le nombre numéro 1 "

Lire a

max \leftarrow a

```
pmax ← 1
POUR i = 2 A 20
  Afficher "Entrez le nombre numéro ", i
  Lire a
  SI a > max ALORS
    max ← a
    pmax ← i
  FIN SI
FIN POUR
Afficher " Le plus grand nombre est : ", max
Afficher " Sa position est : ", pmax
```

5. Le programme est :

```
Entiers : i , a , b , S
Afficher " Entrez un chiffre : "
Lire a
S ← 0
POUR i = 1 A 20
  Afficher " Entrez un nombre : "
  Lire b
  SI a = b ALORS
    S ← S + 1
  FIN SI
FIN POUR
Afficher " Le nombre de fois de saisie de " & a & " est : " & S
```

5.2. La structure TANT QUE

Cette structure permet de répéter les instructions **tant qu'**une condition est satisfaite. Sa syntaxe est :

```
TANT QUE condition
  Instructions à répéter
FIN TANT QUE
```

Condition : c'est une condition qu'on appelle parfois condition d'arrêt. Cette condition est testée avant la première exécution.

Cette structure diffère de la première par le fait qu'on va répéter des instructions pour un nombre de fois inconnu au préalable.

Exemple : Reprenant toujours le programme de notre surveillant. S'il ne sait pas combien de moyennes à calculer on ne pourra pas utiliser la structure **POUR**. Dans ce cas on est obligé d'utiliser la structure **TANT QUE**. Le programme sera alors :

```
Réels : mat, phy, ang, fra, hg, moyenne
Chaîne : reponse
reponse ← "o"
TANT QUE reponse = "o"
  Afficher "Entrez la note de math : "
  Lire mat
  Afficher "Entrez la note de physique : "
  Lire phy
```

```
Afficher "Entrez la note de français :"  
Lire fra  
Afficher "Entrez la note d'anglais :"  
Lire ang  
Afficher "Entrez la note d'histoire-Géo :"  
Lire hg  
moyenne ← ((mat + phy) * 5 + fra * 4 + (ang + hg) * 2) / 18  
Afficher "La moyenne est : ", moyenne  
Afficher "Voulez-vous continuer oui (o) /non (n) ?"  
Lire reponse  
FIN TANT QUE
```

Exercices

1. Ecrire un algorithme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne.
2. Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.
3. Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.
4. Ecrire un algorithme qui demande successivement des nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces nombres et quel était sa position. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.
5. Lire la suite des prix (en dhs entiers et terminée par zéro) des achats d'un client. Calculer la somme qu'il doit, lire la somme qu'il paye, et déterminer le reste à rendre.

Solutions

1. Le programme est :

```
Réel : a  
Tant Que a < 1 OU a > 3  
  Afficher " Veuillez Saisir une valeur comprise entre 1 et 3 "  
  Lire a  
Fin Tant Que
```
2. Le programme est :

```
Réel : a  
Lire a  
Tant Que a < 10 OU a > 20  
  Si a < 10 Alors  
    Afficher " Plus grand ! "  
  Sinon  
    Afficher " Plus petit ! "  
  Fin Si  
Lire a  
Fin Tant Que
```
3. Le programme est :

```
Réel : a , i  
Afficher "Entrez un nombre "
```

```
Lire a
i ← a + 1
Tant Que i < a + 10
    Afficher i
    i ← i + 1
Fin Tant Que
```

4. Le programme est :

```
Entiers : i , a , max , pmax
Afficher " Entrez le nombre numéro 1 "
Lire a
max ← a
pmax ← 1
i ← 1
TANT QUE a <> 0
    i ← i + 1
    Afficher " Entrez le nombre numéro " , i
    Lire a
    SI a > max ALORS
        max ← a
        pmax ← i
    FIN SI
FIN TANT QUE
Afficher " Le plus grand nombre est : " , max
Afficher " Sa position est : " , pmax
```

5. Le programme est :

```
Entiers : prixlu , mdu , mpaye , reste
Afficher " Entrez le prix "
Lire prixlu
mdu ← 0
mdu ← mdu + prixlu
TANT QUE prixlu <> 0
    Afficher " Entrez le prix "
    Lire prixlu
    mdu ← mdu + prixlu
FIN TANT QUE
Afficher " Entrez le prix payé"
Lire mpaye
reste ← mpaye - mdu
afficher " Le reste est : " , reste
```

5.3. La structure FAIRE

Cette structure sert à répéter des instructions **jusqu'à** ce qu'une condition soit réalisée. Sa syntaxe est :

FAIRE

```
Instructions à répéter
JUSQU'A condition
```

Considérons le programme suivant :

```
Entiers : a , c  
FAIRE  
  Lire a  
   $c \leftarrow c * c$   
  Afficher c  
JUSQU'A a = 0
```

Afficher "Fin"

Les mots **FAIRE** et **JUSQU'A** encadrent les instructions à répéter. Cela signifie que ces instructions doivent être répétées autant de fois jusqu'à ce que la variable **a** prenne la valeur 0. Notez bien que le nombre de répétition dans cette structure n'est indiqué explicitement comme c'est le cas de la structure TANT QUE. Il dépend des données que l'on fournit au programme.

Pour bien expliciter cela, voyons ce que produira ce programme si l'on lui fournit successivement les valeurs 2, 4, 0. Le résultat se présentera ainsi :

```
4  
16  
0  
Fin
```

Exercices

1. Ecrire un algorithme qui demande successivement des nombres à l'utilisateur, et qui calcule le nombre de valeurs saisies. La saisie des nombres s'arrête lorsque l'utilisateur entre le caractère « n » ou « N ».
 2. Ecrire un algorithme qui demande successivement des nombres à l'utilisateur, et qui calcule leur moyenne. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.
 3. Modifiez l'algorithme de l'exercice 1, de façon qu'il nous renseigne sur le nombre des valeurs positives et sur le nombre des valeurs négatives. Ne comptez pas les valeurs nuls.
 4. Ecrire un algorithme qui lit les caractères saisis par l'utilisateur. A la fin ce programme nous affichera la phrase saisie. La saisie des caractères s'arrête lorsqu'on tape point « . ». Pour l'utilisateur veut insérer un espace il lui suffit de taper sur 0. Par exemple si l'utilisateur tape successivement les caractères « b », « o », « n », « j », « o », « u », « r », « t », « o », « u », « s », « . », il nous affichera la chaîne « bonjour tous ».
- Mais si il tape « b », « o », « n », « j », « o », « u », « r », « 0 », « t », « o », « u », « s », « . », le programme affichera « bonjour tous ».

Solutions

1. le programme est :

```
Entiers : a , compteur  
Chaîne : reponse  
compteur  $\leftarrow$  0  
FAIRE  
  Afficher " Entrez un nombre : "  
  Lire a  
  compteur  $\leftarrow$  compteur + 1  
  Afficher " Voulez-vous continuer Oui/Non ? "  
  Lire reponse
```

JUSQU'A reponse = " N " ou reponse = "n "
Afficher " Le nombre de valeurs saisies est : ", compteur

2. Le programme est :

Entiers : a , somme , moyenne , compteur
compteur \leftarrow 0

somme \leftarrow 0

FAIRE

Afficher " Entrez un nombre : "

Lire a

compteur \leftarrow compteur + 1

somme \leftarrow somme + a

JUSQU'A a = 0

Moyenne \leftarrow somme / compteur

Afficher " La moyenne de valeurs saisies est : ", moyenne

3. le programme est :

Entiers : a , npos , nneg

Chaîne : reponse

npos \leftarrow 0

nneg \leftarrow 0

FAIRE

Afficher " Entrez un nombre : "

Lire a

SI a > 0 **ALORS**

npos \leftarrow npos + 1

SINON

SI a < 0 **ALORS**

nneg \leftarrow nneg + 1

FIN SI

FIN SI

Afficher " Voulez-vous continuer Oui/Non ? "

Lire reponse

JUSQU'A reponse = " O " ou reponse = " o "

Afficher " Le nombre de valeurs positives saisies est : ", npos

Afficher " Le nombre de valeurs négatives saisies est : ", nneg

4. Le programme est :

Chaînes : caractere , phrase

phrase \leftarrow " "

FAIRE

Afficher "Entrez une caractère : "

Lire caractère

SI caractere = "0" **ALORS**

caractere \leftarrow " "

FIN SI

phrase \leftarrow phrase +caractere

JUSQU'A caractere = " . "

Afficher " La phrase résultante est : " , phrase

6. LES TABLEAUX

6.1. Les tableaux à une seule dimension

Imaginez que l'on veuille calculer la moyenne des notes d'une classe d'élèves. Pour l'instant on pourrait l'algorithme suivant :

```
Réels : somme, nbElevés, Note, i
somme ← 0
Afficher " Nombre d'élèves : "
Lire nbElevés
POUR i = 1 A nbElevés
    Afficher " Note de l'élève numéro ", i , " : "
    Lire Note
    somme ← somme + Note
FIN POUR
```

Afficher " La moyenne est de : " , somme / nbElevés

Si l'on veut toujours calculer la moyenne des notes d'une classe mais en gardant en mémoire toutes les notes des élèves pour d'éventuels calculs (par exemple calculer le nombre d'élèves qui ont des notes supérieures à la moyenne). Dans ce cas il faudrait alors déclarer autant de variables qu'il y a d'étudiants. Donc, si l'on a 10 élèves il faut déclarer 10 variables et si l'on a N il faut déclarer N variables et c'est pas pratique. Ce qu'il faudrait c'est pouvoir par l'intermédiaire d'une seule variable stocker plusieurs valeurs de même type et c'est le rôle des tableaux.

Un **tableau** est un ensemble de valeurs de même type portant le même nom de variable. Chaque valeur du tableau est repérée par un nombre appelé **indice**.

Les tableaux c'est ce que l'on nomme un **type complexe** en opposition aux types de données simples vus précédemment. La déclaration d'un tableau sera via la syntaxe suivante dans la partie des déclarations :

```
Tableau nom_tableau (nombre) : Type
nom_tableau : désigne le nom du tableau
```

nombre : désigne le nombre d'éléments du tableau. On dit aussi sa taille
Type : c'est le type du tableau autrement dit le type de tous ces éléments

Exemples :

Tableau Note (20) : **Réel**

Note (20) est un tableau qui contient vingt valeurs réelles.

Tableau nom (10) , prenom (10) : **Chaîne**

Nom (10) et prenom (10) sont deux tableaux de 10 éléments de type chaîne.

Un tableau peut être représenté graphiquement par (exemple Note (15)) :

Programmation structurée



Si l'on veut accéder (en lecture ou en écriture) à la i ème valeur d'un tableau en utilisant la syntaxe suivante :

nom_tableau (indice)

Par exemple si X est un tableau de 10 entiers :

✚ X (2) ← - 5

met la valeur -5 dans la 3 ème case du tableau

✚ En considérant le cas où a est une variable de type Entier, a ← X (2)

met la valeur de la 3 ème case du tableau tab dans a , c'est-à-dire -5

✚ Lire X (1)

met l'entier saisi par l'utilisateur dans la première case du tableau

✚ Afficher X (0)

affiche la valeur de la première case du tableau

Remarques :

- Un tableau possède un nombre maximal d'éléments défini lors de l'écriture de l'algorithme (les bornes sont des constantes explicites, par exemple 10, ou implicites, par exemple MAX).

Ce nombre d'éléments ne peut être fonction d'une variable.

- La valeur d'un indice doit toujours :

∂ être un nombre entier

∂ être inférieure ou égale au nombre d'éléments du tableau

Exercices

1. Considérons les programmes suivants:

Tableau X (4) : Entier

X (0) ← 12

X (1) ← 5

X (2) ← 8

X (3) ← 20

Tableau voyelle (6) : Chaîne

voyelle (0) ← "a"

voyelle (1) ← "e"

voyelle (2) ← "i"

voyelle (3) ← "o"

voyelle (4) ← "u"

voyelle (5) ← "y"

Donner les représentations graphiques des tableaux X (3) et voyelle (5) après exécution de ces programmes.

2. Quel résultat fournira l'exécution de ce programme :

Entier : i

Tableau C (6) : Entier

POUR i = 0 A 5

Lire C (i)

FIN POUR

POUR i = 0 A 5

C (i) ← C (i) * C (i)

FIN POUR

POUR $i = 0$ **A** 5

Afficher C (i)

FIN POUR

Si on saisit successivement les valeurs : 2 , 5 , 3 , 10 , 4 , 2.

3. Que fournira l'exécution de ce programme :

Tableau suite (8) : **Entier**

Entier : i

Suite (0) \leftarrow 1

Suite (1) \leftarrow 1

POUR $i = 2$ **A** 7

suite (i) \leftarrow suite (i - 1) + suite (i - 2)

FIN POUR

POUR $i = 0$ **A** 7

Afficher suite (i)

FIN POUR

4. Soit T un tableau de vingt éléments de types entiers. Ecrire le programme qui permet de calculer la somme des éléments de ce tableau.

5. Soit T un tableau de N entiers. Ecrire l'algorithme qui détermine le plus grand élément de ce tableau.

6. Ecrire un programme qui permet de lire 100 notes et de déterminer le nombre de celles qui sont supérieures à la moyenne.

7. Soit T un tableau de N entiers. Ecrire l'algorithme qui détermine simultanément la position du plus grand élément et la position du plus petit élément du tableau.

8. Soit T un tableau de N réels. Ecrire le programme qui permet de calculer le nombre des occurrences d'un nombre X (c'est-à-dire combien de fois ce nombre X figure dans le tableau T).

9. On dispose des notes de 25 élèves ; chaque élève peut avoir une ou plusieurs notes mais toujours au moins une. Ecrire un programme permettant d'obtenir la moyenne de chaque élève lorsqu'on lui fournit les notes. On veut que les données et les résultats se présentent ainsi :

| |
|-----------------------------|
| Notes de l'élève numéro 1 |
| 12 |
| 12 |
| -1 |
| Notes de l'élève numéro 2 |
| |
| Notes de l'élève numéro 25 |
| 15 |
| -1 |
| Moyennes |
| Elève numéro 1 : 11 |
| |
| Elève numéro 25 : 15 |
| Moyenne de la classe : 12.3 |

Les parties italiques correspondent aux données tapées par l'utilisateur. La valeur -1 sert de critère de fin de notes pour chaque élève.

Solutions

1. La représentation graphique du tableau X (4) après exécution du premier programme est :

| | | | |
|----|---|---|----|
| 12 | 5 | 8 | 20 |
|----|---|---|----|

La représentation graphique du tableau voyelle (4) après exécution du deuxième programme est :

| | | | | | |
|---|---|---|---|---|---|
| a | e | i | o | u | y |
|---|---|---|---|---|---|

2. L'exécution du programme nous affichera successivement à l'écran :

| |
|-----|
| 4 |
| 25 |
| 9 |
| 100 |
| 16 |
| 4 |

3. L'exécution du programme nous affichera successivement à l'écran :

| |
|-------|
| 1 |
| 12358 |
| 13 |
| 21 |

4. Le programme est :

```
Entiers i , somme
Tableau T (N) : ENTIER
somme ← 0
POUR i = 0 A N-1
    somme ← somme + T (i)
FIN POUR
```

Afficher " La somme de tous les éléments du tableau est : " , somme

5. Le programme est :

```
ENTIERS : i , max
Tableau T (N) : ENTIER
max ← T (0)
i ← 0
FAIRE
    i ← i + 1
    SI T (i) > max ALORS
        max ← T (i)
    FIN SI
JUSUQ'A i = N
```

Afficher " La somme de tous les éléments du tableau est : " , somme

6. Le programme est :

```
Réels i , somme , moyenne , nsup
Tableau Note (100) : Réel
somme ← 0
POUR i = 0 A 99
    Lire Note (i)
    somme ← somme + Note (i)
FIN POUR
Moyenne ← somme / 100
nsup ← 0
POUR i = 0 A 99
    SI Note (i) > moyenne ALORS
        nsup ← nsup + 1
    FIN SI
FIN POUR
```

Afficher " Le nombre de notes supérieures à la moyenne est : " , nsup

7. Le programme est :

```
Entiers i , pmax , pmin
Tableau T (N) : Réel
max ← T (0)
min ← T (0)
pmax ← 1
pmin ← 1
i ← 1
FAIRE
    i ← i + 1
    SI T (i) > max ALORS
        max ← T (i)
        pmax ← i
    FIN SI

    SI T (i) < min ALORS
        min ← T (i)
        pmin ← i
    FIN SI
JUSQU'À i = N
```

Afficher " La position du plus grand élément du tableau est : " , pmax

Afficher " La position du plus petit élément du tableau est : " , pmin

8. Le programme est :

```
Réels : X ,i,Compt
Entier Compt
Tableau T (N) : Réel
Lire X
POUR i=0 A i=N-1
    SI T (i) =X ALORS
```

Compt ← compt+1

FIN SI

FIN POUR

Afficher " Le nombre d'occurrences de cet éléments du tableau est : " , compt

9. Le programme est :

Entiers : i , note , nnote , snote , smoyenne , cmoyenne

Tableau moy (25) : **Réel**

POUR i = 1 A 25

Afficher " Notes de l'élève numéro " , i

snote ← 0

nnote ← 0

FAIRE

Lire note

SI note < > -1 **ALORS**

snote ← snote + note

nnote ← nnote + 1

FIN SI

JUSQU'A note = -1

moy (i) = snote / nnote

smoyenne = smoyenne + moy (i)

FIN POUR

Afficher " Moyennes "

POUR i = 0 A 24

Afficher " Elève numéro " , i , " : " , moy (i)

FIN POUR

cmoyenne = smoyenne / 25

afficher " Moyenne de la classe : " , cmoyenne

6.2. Les tableaux dynamiques

Il arrive fréquemment que l'on ne connaisse pas à l'avance le nombre d'éléments que devra comporter un tableau. Bien sûr, une solution consisterait à déclarer un tableau avec une taille très grande. Cela pourra avoir comme conséquence soit que cette taille ne nous suffira pas ou qu'une place mémoire immense sera réservée sans être utilisée.

Afin de surmonter ce problème on a la possibilité de déclarer le tableau sans préciser au départ son nombre d'éléments. Ce n'est que dans un second temps, au cours du programme, que l'on va fixer ce nombre via une instruction de re-dimensionnement : **Redim**.

Exemple :

On veut saisir des notes pour un calcul de moyenne, mais on ne sait pas combien il y aura de notes à saisir. Le début de l'algorithme sera quelque chose du genre :

Tableau Notes () : **Réel**

Entier nb

Afficher "Combien y a-t-il de notes à saisir ?"

Lire nb

Redim Notes(nb-1)

...

Exercices

1. Insertion d'un élément dans un tableau

Soit T un tableau de N éléments. Ecrire un programme qui permet d'insérer un élément x à la position i du tableau T.

2. Suppression d'un élément du tableau

Soit T un tableau de N éléments. Ecrire un programme qui permet de supprimer un élément x du tableau.

Solutions

1. Le programme est :

```
Tableau T () : Entier
Entier i, x, j
Afficher " Donnez la dimension du tableau "
Lire N
Redim T (N)
POUR j = 0 A N-1
  Lire T (j)
FIN POUR
Afficher " Entrez le nombre à insérer "
Lire x
Afficher " Entrez la position où insérer ce nombre "
Lire i
Redim T (N +1)
j = N
TANT QUE j >= i
  T (j+1) = T (j)
  j = j - 1
FIN TANT QUE
T (i) = x
```

Dans ce programme on a travaillé avec un seul tableau dynamique. On peut aussi travailler avec le tableau T à dimension fixe et en définir un autre qui recevra tous les éléments de T plus l'élément à insérer. Le programme dans ce cas est :

```
Tableau T (N) : Entier
Tableau Tr (N+1) : Entier
Entier : i , x , j , k
POUR j = 0 A N-1
  Lire T (j)
FIN POUR
Afficher " Entrez le nombre à insérer "
Lire x
Afficher " Entrez la position où insérer ce nombre "
Lire i
j = 1
k = 1
TANT QUE k <= N + 1
  SI k <> i ALORS
    Tr (k) ← T (j)
```

```

        j ← j + 1
    SINON
        Tr (k) = x
    FIN SI
    k = k + 1
FIN TANT QUE
    
```

2. Le programme est :

```

Tableau T (N) : Entier
Tableau Tr () : Entier
Entiers : i , x , j
POUR j = 0 A N-1
    Lire T (j)
FIN POUR
Afficher " Entrez le nombre à supprimer "
Lire x
j ← 0
    POUR i = 0 A N-1
        SI T (i) <> x ALORS
            j ← j + 1
            ReDim Tr (j)
            Tr (j) = T (i)
        FIN SI
    FIN POUR
    
```

Dans ce programme on a considéré deux tableaux, le tableau T à dimension fixe et le tableau Tr dynamique. Il est aussi possible de travailler avec un seul tableau dynamique.

```

Tableau T () : Entier
Entiers : x, j , k , N
Afficher " Donnez la dimension du tableau "
Lire N
Redim T (N)
POUR j = 0 A N-1
    Lire T (j)
FIN POUR
Afficher " Entrez le nombre à supprimer "
Lire x
j = 0
    TANT QUE j <= N
        SI T (j) = x ALORS
            POUR k = j A N - 1
                T (k) = T (k + 1)
            FIN POUR
            N ← N - 1
            ReDim T (N)
        SINON
            j ← j + 1
        FIN SI
    FIN TANT QUE
    
```

6.3. Les tableaux multidimensionnels(Matrice)

Nous avons vu qu'un tableau à une dimension correspond à une liste ordonnée de valeurs, repérée chacune par un indice.

Dans tous les langages de programmation, il est possible de définir des tableaux à deux dimensions (permettant par exemple de représenter des matrices). Ainsi, on pourra placer des valeurs dans un tableau à deux dimensions et cela consiste comme dans le cas des tableaux à une dimension à donner un nom à l'ensemble de ces valeurs. Chaque valeur est alors repérée par deux indices qui précise la position.

On déclare un tableau à deux dimensions de la façon suivante :

Tableau *nom_tableau* (*i* , *j*) : **Type**
nom_tableau : désigne le nom du tableau

i : désigne le nombre de lignes du tableau

j : désigne le nombre de colonnes du tableau

Type : représente le type des éléments du tableau

Exemple :

Soit T (3 , 5) un tableau d'entiers. On peut représenter graphiquement par :

| | | | | |
|--|--|--|--|--|
| | | | | |
| | | | | |
| | | | | |

T (0 , 3) →

← T (2,3)

T (0 ,3) et T(2 , 4) sont deux éléments du tableau. Entre parenthèse on trouve les valeurs des indices séparées par une virgule. Le premier sert à repérer le numéro de la ligne, le second le numéro de la colonne.

On accède en lecture ou en écriture à la valeur d'un élément d'un tableau à deux dimensions en utilisant la syntaxe suivante :

Nom_tableau (*i* , *j*)

Par exemple si T est défini par : **Tableau** T (3 , 2) : Réel

∂ T (2 , 1) ← -1.2

met la valeur -1.2 dans la case 2,1 du tableau

∂ En considérant le cas où a est une variable de type Réel, a ← T (2 , 1)

met -1.2 dans a .

Par extension, on peut aussi définir des tableaux à n dimensions. Leur déclaration sera à l'image de celle des tableaux à deux dimensions, c'est-à-dire :

Tableau *nom_tableau* (*i* , *j* , *k* ,) : **Type**

Par exemple :

Tableau X (10 , 9 , 5) : **Entier**

Ainsi que leur utilisation :

- X (2 , 1 , 3) ← 10
- a ← X (2 , 1 , 3)
-

Exercices

1. Considérons le programme suivant :

Tableau X (2 , 3) : **Entier**

Entiers : i , j , val

```
val ← 1
POUR i = 0 A 1
    POUR j = 0 A 2
        X (i , j) ← val
        val ← val + 1
    FIN POUR
FIN POUR
POUR i = 0 A 1
    POUR j = 0 A 2
        Afficher X (i , j)
    FIN POUR
FIN POUR
```

- a. Que produit l'exécution de ce programme.
b. que produira ce programme si l'on remplace les dernières lignes par :

```
POUR j = 0 A 2
    POUR i = 0 A 1
        Afficher X (i , j)
    FIN POUR
FIN POUR
```

2. Quel résultat fournira ce programme :

```
Tableau X (4 , 2) : Entier
Entiers : k , m
POUR k = 0 A 3
    POUR m = 0 A 1
        X (k , m) ← k + m
    FIN POUR
FIN POUR
POUR k = 0 A 3
    POUR m = 0 A 1
        Afficher X (k , m)
    FIN POUR
FIN POUR
```

3. Soit T un tableau à deux dimensions de vingt lignes et cinquante colonnes.

- a. Ecrire un algorithme qui permet de calculer la somme de tous les éléments du tableau.
b. Ecrire l'algorithme qui permet de compter le nombre des éléments strictement positifs.
c. Ecrire l'algorithme permettant d'obtenir la somme des éléments positifs (spos) et la somme des éléments négatifs (sneg) de ce tableau.
d. Ecrire l'algorithme qui détermine la plus grande valeur des éléments du tableau.
e. Ecrire l'algorithme qui détermine simultanément l'élément le plus grand du tableau ainsi que sa position.

Solutions

1. L'exécution du programme donnera :
Les deux premières boucles du programme permettent de remplir le tableau.

Programmation structurée

Ainsi la représentation graphique sera :

| | | |
|---|---|---|
| 0 | 1 | 2 |
| 3 | 4 | 5 |

Les deux dernières boucles permettent d'afficher ces six éléments. Le résultat sera donc :

Question a

0
1
2
3
4
5

Question b

0
3
1
4
2
5

2. Les deux premières boucles de ce programme permettent de remplir le tableau. La représentation graphique sera ainsi :

| | |
|---|---|
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |

Les dernières boucles permettent d'afficher les huit éléments du tableau. Le résultat est :

0
1
1
2
2
3
3
4

3. Soit T (20 , 50) un tableau de réels.

a. L'algorithme qui calcule la somme de tous les éléments du tableau est :

Tableau T (20 , 50) : Réel

Entiers i , j : Entiers

Réel : som

som \leftarrow 0

POUR i = 0 **A** 19

```
POUR j = 0 A 49
    som ← som + T (i , j)
FIN POUR
```

FIN POUR

Afficher ” La somme de tous les éléments du tableau est : ” , som

b. L’algorithme qui compte le nombre des éléments strictement positifs est :

```
Tableau T (20 , 50) : Réel
Entiers i , j
Réel : npos
npos ← 0
POUR i = 0 A 19
    POUR j = 0 A 49
        SI T (i , j) > 0 ALORS
            npos ← npos + 1
        FIN SI
    FIN POUR
FIN POUR
```

FIN POUR

FIN POUR

Afficher ” Le nombre des éléments strictement positifs du tableau est : ” , npos

c. L’algorithme permettant d’obtenir la somme des éléments positifs (spos) et la somme des éléments négatifs (sneg) de ce tableau est :

```
Tableau T (20 , 50) : Réel
Entiers : i , j
Réel spos , sneg
spos ← 0
sneg ← 0
POUR i = 0 A 19
    POUR j = 0 A 49
        SI T (i , j) > 0 ALORS
            spos ← spos + T (i , j)
        SINON
            sneg ← sneg + T (i , j)
        FIN SI
    FIN POUR
FIN POUR
```

FIN POUR

FIN POUR

Afficher ” La somme des éléments positifs du tableau est : ” , spos

Afficher ” La somme des éléments négatifs du tableau est : ” , sneg

d. L’algorithme qui détermine la plus grande valeur des éléments du tableau est :

```
Tableau T (20 , 50) : Réel
Entiers : i , j
Réel : max
max ← T (1 , 1)
POUR i = 0 A 19
    POUR j = 0 A 49
        SI T (i , j) > max ALORS
            max ← T (i , j)
        FIN SI
    FIN POUR
FIN POUR
```

FIN POUR

Afficher ” Le plus grand élément du tableau est : ‘’, max

Afficher ” la position de l’élément i ‘’ = ‘’, imax, ‘’ et j= ‘’, jmax

e. L’algorithme qui détermine simultanément l’élément le plus grand du tableau ainsi que sa position est :

```
Tableau T (20 , 50) : Réel  
Entier i , j , imax , jmax  
Réel : max  
max ← T (1 , 1)  
POUR i = 0 A 19  
    POUR j = 0 A 49  
        SI T (i , j) > max ALORS  
            max ← T (i , j)  
            imax ← i  
            jmax ← j  
        FIN SI  
    FIN POUR  
FIN POUR
```

Afficher ” Le plus grand élément du tableau est : ‘’, max

7. LES STRUCTURES

Imaginons que l’on veuille afficher les notes d’une classe d’élèves par ordre croissant avec les noms et prénoms de chaque élève. On va donc utiliser trois tableaux (pour stocker les noms, les prénoms et les notes). Lorsque l’on va trier le tableau des notes il faut aussi modifier l’ordre les tableaux qui contiennent les noms et prénoms. Mais cela multiplie le risque d’erreur. Il serait donc intéressant d’utiliser ce qu’on appelle **les structures**.

Les structures contrairement aux tableaux servent à rassembler au sein d’une seule entité un ensemble fini d’éléments de type éventuellement différents. C’est le deuxième type complexe disponible en algorithmique.

A la différence des tableaux, il n’existe pas par défaut de type structure c’est-à-dire qu’on ne peut pas déclarer une variable de type structure. Ce qu’on peut faire c’est de construire toujours un nouveau type basé sur une structure et après on déclare des variables sur ce nouveau type. La syntaxe de construction d’un type basé sur une structure est :

```
TYPE NomDuType = STRUCTURE  
    attribut1 : Type  
    attribut2 : Type  
    ...  
    attributn : Type  
FIN STRUCTURE
```

Le type d’un attribut peut être :

- ✓ Un type simple
- ✓ Un type complexe
 - Un tableau
 - Un type basé sur une structure

Exemple :

```
TYPE Etudiant = STRUCTURE  
    nom : chaîne  
    prenom : chaîne  
    note : Réel  
FIN STRUCTURE
```

Dans cet exemple on a construit un type *Etudiant* basé sur une structure. Cette structure a trois attributs (on dit aussi champ) : nom, prenom et note.

```
TYPE Date = STRUCTURE  
jour : Entier  
mois : Entier  
annee : Entier  
FIN STRUCTURE
```

Dans ce deuxième exemple, on a construit un type *Date* basé sur une structure. Cette structure a aussi trois attributs : jour, mois et annee.

Après on peut déclarer des variables basé sur ce type. Par exemple :

Etudiant : Etud

Donc Etud est une variable de type Etudiant.

Il est possible de déclarer un tableau d'éléments de ce type Etudiant par exemple. On pourra écrire donc :

Tableau Etud (20) : Etudiant

Etud (0) représente le premier étudiant.

Maintenant, pour accéder aux attributs d'une variable dont le type est basé sur une structure on suffixe le nom de la variable d'un point « . » suivi du nom de l'attribut. Par exemple, dans notre cas pour affecter le nom "Dinar" à notre premier étudiant, on utilisera le code suivant :

Etud (0).nom = "Dianr"

Exercices

1. Définissez la structure « Stagiaire » constituée des champs suivants :

| Champ | Type |
|----------|-----------|
| Nom | Chaîne |
| Prénom | Chaîne |
| Détenais | Structure |

Le champ « Détenais » est aussi une structure dont les champs sont :

| Champ | Type |
|-------|--------|
| Jour | Entier |
| Mois | Entier |
| Année | Entier |

Ecrivez ensuite l'algorithme qui permet de lire et après afficher le nom, prénom et date de naissance d'un stagiaire.

2. On souhaite gérer les notes d'un étudiant. Pour cela on va définir la structure « Etudiant » dont les champs sont :

| Champ | Type |
|---------|-----------------------|
| Nom | Chaîne |
| Prénom | Chaîne |
| Note | Tableau de 3 éléments |
| Moyenne | Réel |

Ecrire l'algorithme qui permet de lire les informations d'un étudiant (nom, prénom et notes), de calculer sa moyenne et d'afficher à la fin un message sous la forme suivante : « La moyenne de l'étudiant Dinar Youssef est : 12.45 » où « Dinar » et « Youssef » sont les noms et prénoms lus et 12.45 est la moyenne calculée.

3. Modifier l'algorithme de l'exercice précédent de façon que l'on puisse gérer les notes de 50 étudiants.

Solutions

1. L'algorithme est :

TYPE Date = STRUCTURE

Jour : **Entier**

Mois : **Entier**

Annee : **Entier**

FIN STRUCTURE

TYPE Stagiaire = STRUCTURE

Nom : **chaîne**

Prenom : **chaîne**

Datenais : **Date**

FIN STRUCTURE

Stagiaire : stag

Afficher " Entrez les information du stagiaire "

Afficher " Entrez le nom "

Lire stag.Nom

Afficher " Entrez le prénom "

Lire stag.Prenom

Afficher " Entrez le jour de naissance "

Lire stag.Date.Jour

Afficher " Entrez le mois de naissance "

Lire stag.Date.Mois

Afficher " Entrez l'année de naissance "

Lire stag.Date.Annee

Afficher " Le nom du stagiaire est : " , stag.Nom

Afficher " Son prénom est : " , stag.Prenom

Afficher " Sa date de naissance est : " , stag.Date.Jour , "/" , stag.Date.Mois, "/" , stag.Date.Annee

2. L'algorithme est :

TYPE Etudiant = STRUCTURE

Nom : **Chaîne**

Prenom : **Chaîne**

Note (3) : **Réel**

Moyenne : **Réel**

FIN STRUCTURE

Entier : i

Réel : som

Etudiant : etud

Afficher " Entrez les information de l'étudiant "

Afficher " Entrez le nom "

Lire etud.Nom

Afficher " Entrez le prénom "

Lire etud.Prenom

Afficher " Entrez la première note "

Lire Etud.Note (1)

Afficher " Entrez la deuxième note "

Lire etud.Note (2)

Afficher " Entrez la troisième note "

Lire etud.Note (3)

som \leftarrow 0

POUR i = 1 A 3

 som \leftarrow etud.Note (i)

FIN POUR

etud.Moyenne = som / 3

Afficher " La moyenne de l'étudiant " , etud.Nom , " " , etud.Prenom , " est : " , etud.Moyenne

3. L'algorithme est :

TYPE Etudiant = STRUCTURE

Nom : **Chaîne**

Prenom : **Chaîne**

Note(3) : **Réel**

Moyenne : **Réel**

FIN STRUCTURE

Entier i, j

Réel : som

Etudiant : etud(50)

Afficher " Entrez les information des étudiants "

POUR j = 0 A 49

Afficher " Entrez le nom "

Lire etud(j).Nom

Afficher " Entrez le prénom "

Lire etud(j).Prenom

Afficher " Entrez la première note "

```
    Lire etud(j).Note (1)
    Afficher ” Entrez la deuxième note “
    Lire etud(j).Note (2)
    Afficher ” Entrez la troisième note “
    Lire etud(j).Note (3)
    som ← 0
    POUR i = 0 A 2
        som ← etud(j).Note (i)
    FIN POUR
    etud (j).Moyenne = som / 3
FIN POUR
POUR j = 1 A 50
    Afficher ” La moyenne de l’étudiant “ , etud(j).Nom , “ ” , etud(j).Prenom , “ est : “ ,
    etud(j).Moyenne
FIN POUR
```

8. LES FONCTIONS ET PROCEDURES

En programmation, donc en algorithmique, il est possible de décomposer le programme qui résout un problème en des sous-programmes qui résolvent des sous parties du problème initial. Ceci permettra d’améliorer la conception du programme et ainsi sa lisibilité.

L’utilisation des sous-programmes s’avère utile aussi dans le cas où on constate qu’une suite d’actions se répète plusieurs fois.

Il existe deux types de sous-programmes les fonctions et les procédures. Un sous- programme est obligatoirement caractérisé par un nom (un identifiant) unique.

Le nom d’un sous-programme comme le nom d’une variable doit :

- ∂ Contenir que des lettres et des chiffres
- ∂ Commencer obligatoirement par une lettre

Le programme qui utilise un sous- programme est appelé **programme appelant**. Un sous-programme peut être invoqué n’importe où dans le programme appelant en faisant référence à son nom.

Un programme ainsi doit suivre la structure suivante :

Définition des constantes

Définition des types

Déclaration des variables

Définition des sous- programmes

8.1. Les fonctions

Une fonction est un sous-programme qui retourne un **seul** résultat. Pour définir une fonction on utilise la syntaxe suivante :

```
FONCTION nom_fonction (Argument1 : Type , Argument2 : Type , ...) : Type
    Déclarations
    Instructions de la fonction
    nom_fonction ← Valeur renvoyée
FIN DE FONCTION
```

On constate que la déclaration d’une fonction revient à lui préciser un nom, un type ainsi qu’une liste d’arguments.

Un argument (appelé **paramètre formel**) d'un sous-programme est une variable locale particulière qui est associée à une variable ou constante du programme appelant. Puisque qu'un argument est une variable locale, il admet un type.

Lorsque le programme appelant appelle le sous-programme il doit indiquer la variable ou la constante de même type, qui est associée au paramètre.

Par exemple, le sous-programme *sqr* permet de calculer la racine carrée d'un réel. Ce sous-programme admet un seul paramètre de type réel positif.

Le programme qui utilise *sqr* doit donner le réel positif dont il veut calculer la racine carrée, cela peut être :

- une variable, par exemple *a*
- une constante, par exemple 5.25

Les arguments d'une fonction sont en nombre fixe (≥ 0).

Une fonction possède un seul type, qui est le type de la valeur retournée qui est affecté au nom de la fonction.

Une fois la fonction définie, il est possible (en fonction des besoins) à tout endroit du programme appelant de faire appel à elle en utilisant simplement son nom suivi des arguments entre parenthèses.

Les parenthèses sont toujours présentes même lorsqu'il n'y a pas de paramètre.

Les arguments spécifiés lors de l'appel de la fonction doivent être en même nombre que dans la déclaration de la fonction et des types prévus. Dans le programme appelant ces arguments sont appelés **paramètres effectifs**.

La valeur ainsi renvoyée par la fonction peut être utilisée dans n'importe quelle expression compatible avec son type.

Exemple :

```
FONCTION Calcul (x : Réel , y : Réel , z : Réel) : Réel
Entier a
a ← 3
Calcul ← (x + y + z) * a
Fin fonction
```

Ça c'est un exemple de déclaration de fonction. Cette fonction appelée « Calcul » est de type réel et elle admet trois arguments de type réel.

Maintenant voici un exemple de programme complet :

```
FONCTION Calcul (x : Réel , y : Réel , z : Réel) : Réel
Entier : a
a ← 3
Calcul ← (x + y + z) * a
FIN fonction
Réels : i , j , k , b
Lire i
Lire j
Lire k
b ← Calcul (i , j , k) + 2
Afficher b
```

Dans ce programme on fait appel à une fonction. Sa valeur est utilisée dans une expression.

Exercice

1. Définir la fonction « Somme » qu'on lui passe deux valeurs de type entier et qui renvoie comme valeur la somme des valeurs reçues.

2. Définir la fonction « Absolue » qui renvoie la valeur absolue d'une valeur qu'on lui passe comme paramètre.
3. Définir la fonction « Inverse » qui renvoie l'inverse d'une valeur qu'on lui passe comme paramètre.
4. Définir la fonction « Max » qui renvoie le maximum de deux valeurs.
5. Ecrivez un programme qui lit trois scores et qui utilise la fonction définie dans l'exercice précédent pour déterminer le meilleur score et l'afficher après.

Solutions

1. La définition de la fonction « Somme » est :

```
FUNCTION Somme (x : Réel , y : Réel ) : Réel
Somme ← x + y
FIN fonction
```

2. La définition de la fonction « Absolue » est :

```
FUNCTION Absolue (x : Réel) : Réel
SI x > 0 ALORS
    Absolue ← x
SINON
    Absolue ← -1 * x
FIN SI
FIN FONCTION
```

3. La définition de la fonction « Inverse » est :

```
FUNCTION Inverse (x : Réel) : Réel
SI x ≠ 0 ALORS
    Inverse ← 1 /x
FIN SI
FIN FONCTION
```

4. La définition de la fonction « Max » est :

```
FUNCTION Max (x : Réel , y Réel) : Réel
SI x > y ALORS
    Max ← x
SINON
    Max ← y
FIN SI
FIN FONCTION
```

5. Le programme est :

```
FUNCTION Max (x : Réel , y Réel) : Réel
SI x > y ALORS
    Max ← x
SINON
    Max ← y
FIN SI
FIN FONCTION
Réels : score1 , score2 , score3 , meil_score
Afficher ” Entrez les trois scores : ‘’
```

Lire score1

Lire score2

Lire score3

meil_score \leftarrow Max (Max (score1 , score2) , score3)

Afficher "Le meilleur score est : " , meil_score

8.2. Les variables locales et globales

La **portée** d'une variable est l'ensemble des sous- programmes où cette variable est connue c'est-à-dire que les instructions de ces sous-programmes peuvent utiliser cette variable.

Une variable définie au niveau du programme principal (problème appelant) est appelée **variable globale**. Sa portée est totale : **tout** sous-programme du programme principal peut utiliser cette variable.

Cependant, une variable définie au sein d'un sous-programme est appelée **variable locale**. La portée d'une variable locale est uniquement le sous-programme qui la déclare.

Remarque :

Lorsque le nom d'une variable locale est identique à une variable globale, la variable globale est localement masquée. Dans ce sous-programme la variable globale devient inaccessible.

Exemple

Soit le programme suivant :

Fonction Surface (a : **Réel**) : **Réel**

Réels : valeur , resultat

valeur \leftarrow 3.14

resultat \leftarrow valeur * a

Surface \leftarrow resultat

FIN FONCTION

Réel : rayon

Afficher " Entrez le rayon du cercle : "

Lire rayon

Afficher " La surface de cette cercle est : " , Surface (rayon)

Les variables *valeur* et *resultat* déclarées dans la fonction *Surface* sont locales.

Considérons presque le même programme sauf que la variable *valeur* est déclarée maintenant dans le programme appelant.

Fonction Surface (a : **Réel**) : **Réel**

Réels : resultat

resultat \leftarrow valeur * a

Surface \leftarrow resultat

FIN FONCTION

Variable valeur , rayon : **Réel**

valeur \leftarrow 3.14

Afficher "Entrez le rayon du cercle : "

Lire rayon

Afficher " La surface de cette cercle est : " , Surface (rayon)

Dans ce deuxième programme seule la variable *resultat* est locale. Tandis que la variable *valeur* est devenue globale. On peut par conséquent accéder à sa valeur dans la fonction *Surface*.

8.3. Les passage de paramètres

Il existe deux types d'association (que l'on nomme **passage de paramètre**) entre le(s) paramètre(s) du sous-programme (fonction ou procédure) et variable(s) du programme appelant :

- **Passage par valeur**
- **Passage par référence(ou par adresse)**

Dans le cas où l'on choisit pour un paramètre effectif un passage par valeur, la valeur de ce paramètre effectif ne change pas même si lors de l'appel du sous-programme la valeur du paramètre formel correspondant change. On peut dire que dans ce cas le paramètre effectif et le paramètre formel ont font deux variables différents qui ont seulement la même valeur. C'est la type de passage par défaut. Dans le cas où l'on choisit pour un paramètre effectif un passage par adresse, la valeur de ce paramètre effectif change si lors de l'appel du sous-programme la valeur du paramètre formel correspondant change.

On peut dire que dans ce cas le paramètre effectif et le paramètre formel ont font deux variables qui ont le même adresse (par conséquent valeur). Pour préciser qu'il s'agit d'un passage par adresse, il faut soulignés les paramètres concernés lors de la définition du sous-programme.

Exemple

Considérons les deux programmes suivants :

Programme 1

Fonction Calcul (a : **Réel**) : **Réel**

Calcul \leftarrow a * 2

a \leftarrow a - 1

FIN FONCTION

Réel : x

x \leftarrow 3

Afficher Calcul (x)

Afficher x

Programme 2

Fonction Calcul (a : **Réel**) : **Réel**

Calcul \leftarrow a * 2

a \leftarrow a - 1

FIN FONCTION

Réel : x

x \leftarrow 3

Afficher Calcul (x)

Afficher x

Dans le premier programme on a un passage de paramètre par valeur et dans le deuxième on a un passage de paramètres par référence . Le premier programme affichera le résultat suivant : 63

car même si la valeur de *a* change celle de *x* non.

Tandis que le deuxième programme il affichera : 62

la valeur de *x* changera car celle de *a* a changée.

8.4. Les procédures

Les procédures sont des sous- programmes qui ne retournent **aucun** résultat. Elles admettent comme les fonctions des paramètres.

On déclare une procédure de la façon suivante :

PROCEDURE *nom_procedure* (*Argument1* : **Type** , *Argument2* : **Type** ,)
 Déclarations
 Instructions de la procédure
FIN PROCEDURE

Et on appelle une procédure comme une fonction, en indiquant son nom suivi des paramètres entre parenthèses.

Exercice

1. Ecrire une procédure qui reçoit la longueur et la largeur d'une surface et qui affiche la valeur de la surface. Donnez à cette procédure le nom « Surface ».
2. Ecrire une procédure qui permet d'échanger les valeurs de deux variables. Appelez cette procédure « Echanger ».
3. On dispose d'une phrase dont les mots sont séparés par des points virgules. Ecrivez une procédure qui permet de remplacer les points virgules par des espaces. On suppose qu'on dispose des fonctions suivantes :

Calcul \leftarrow a * 2
a \leftarrow a - 1

Réel : x
x \leftarrow 3

Afficher Calcul (x)
Afficher x

Programme 2

Fonction Calcul (a : Réel) : Réel
Calcul \leftarrow a * 2
a \leftarrow a - 1
FIN FONCTION

Réel : x
x \leftarrow 3
Afficher Calcul (x)
Afficher x

Dans le premier programme on a un passage de paramètre par valeur et dans le deuxième on a un passage de paramètres par référence (par adresse). Le premier programme affichera le résultat suivant : 63

car même si la valeur de *a* change celle de *x* non.

Tandis que le deuxième programme il affichera : 62
la valeur de *x* changera car celle de *a* a changée.

9. LES ALGORITHMES DE TRI

Trier les éléments d'un tableau revient à ordonner tous ces éléments selon un ordre croissant ou décroissant.

Soit T un tableau de N éléments muni d'une relation d'ordre \leq . Trier ce tableau c'est construire un algorithme qui devra satisfaire à la spécification suivante :

i [1 , N-1] T (i) \leq T (i+1)

Dans ce paragraphe on va traiter plusieurs algorithmes de tri : tri par sélection, tri par bulle, tri par comptage, tri par insertion, tri par shell.
(voir la partie exercices).

10. LES FICHIERS

« On ne peut pas davantage créer des fichiers numériques non copiables que créer de l'eau non humide » - Bruce Schneider

Jusqu'à présent, les informations utilisées dans nos programmes ne pouvaient provenir que de deux sources : soit elles étaient incluses dans l'algorithme lui-même, par le programmeur, soit elles étaient entrées en cours de route par l'utilisateur. Mais évidemment, cela ne suffit pas à combler les besoins réels des informaticiens.

Imaginons que l'on veuille écrire un programme gérant un carnet d'adresses. D'une exécution du programme à l'autre, l'utilisateur doit pouvoir retrouver son carnet à jour, avec les modifications qu'il y a apportées la dernière fois qu'il a exécuté le programme. Les données du carnet d'adresse ne peuvent donc être incluses dans l'algorithme, et encore moins être entrées au clavier à chaque nouvelle exécution !

Les fichiers sont là pour combler ce manque. **Ils servent à stocker des informations de manière permanente, entre deux exécutions d'un programme.** Car si les variables, qui sont je le rappelle des adresses de mémoire vive, disparaissent à chaque fin d'exécution, les fichiers, eux sont stockés sur des périphériques à mémoire de masse (disquette, disque dur, CD Rom...).

10.1. TYPES D'ACCES

On vient de voir que l'organisation des données au sein des enregistrements du fichier pouvait s'effectuer selon deux grands choix stratégiques. Mais il existe une autre ligne de partage des fichiers : le type d'accès, autrement dit la manière dont la machine va pouvoir aller rechercher les informations contenues dans le fichier.

On distingue :

- **L'accès séquentiel** : on ne peut accéder qu'à la donnée suivant celle qu'on vient de lire. On ne peut donc accéder à une information qu'en ayant au préalable examiné celle qui la précède. Dans le cas d'un fichier texte, cela signifie qu'on lit le fichier ligne par ligne (enregistrement par enregistrement).
- **L'accès direct (ou aléatoire)** : on peut accéder directement à l'enregistrement de son choix, en précisant le numéro de cet enregistrement. Mais cela veut souvent dire une gestion fastidieuse des déplacements dans le fichier.
- **L'accès indexé** : pour simplifier, il combine la rapidité de l'accès direct et la simplicité de l'accès séquentiel (en restant toutefois plus compliqué). Il est particulièrement adapté au traitement des gros fichiers, comme les bases de données importantes.

A la différence de la précédente, cette typologie ne caractérise pas la structure elle-même du fichier. En fait, tout fichier peut être utilisé avec l'un ou l'autre des trois types d'accès. Le choix du type d'accès n'est pas un choix qui concerne le fichier lui-même, mais uniquement la manière dont il va être traité par la machine. *C'est donc dans le programme, et seulement dans le programme, que l'on choisit le type d'accès souhaité.*

10.2. INSTRUCTIONS (fichiers texte en accès séquentiel)

Si l'on veut travailler sur un fichier, la première chose à faire est de l'**ouvrir**. Cela se fait en attribuant au fichier un numéro de canal. On ne peut ouvrir qu'un seul fichier par canal, mais quel que soit le langage, on dispose toujours de plusieurs canaux, donc pas de soucis.

L'important est que lorsqu'on ouvre un fichier, on stipule ce qu'on va en faire : *lire, écrire ou ajouter*.

- Si on ouvre un fichier **pour lecture**, on pourra uniquement récupérer les informations qu'il contient, sans les modifier en aucune manière.
- Si on ouvre un fichier **pour écriture**, on pourra mettre dedans toutes les informations que l'on veut. Mais les informations précédentes, si elles existent, seront *intégralement écrasées*. Et on ne pourra pas accéder aux informations qui existaient précédemment.
- Si on ouvre un fichier **pour ajout**, on ne peut ni lire, ni modifier les informations existantes. Mais on pourra, comme vous commencez à vous en douter, ajouter de nouvelles lignes (je rappelle qu'au terme de lignes, on préférera celui d'**enregistrements**).

Au premier abord, ces limitations peuvent sembler infernales. Au deuxième abord, elles le sont effectivement. Il n'y a même pas d'instructions qui permettent de supprimer un enregistrement d'un fichier !

Toutefois, avec un peu d'habitude, on se rend compte que malgré tout, même si ce n'est pas toujours marrant, on peut quand même faire tout ce qu'on veut avec ces fichiers séquentiels. Pour ouvrir un fichier texte, on écrira par exemple :

Ouvrir "Exemple.txt" sur 4 en Lecture

Ici, "Exemple.txt" est le nom du fichier sur le disque dur, 4 est le numéro de canal, et ce fichier a donc été ouvert en lecture. Vous l'aviez sans doute pressenti. Allons plus loin :

Caractères : Truc, Nom, Prénom, Tel, Mail

Ouvrir "Exemple.txt" sur 4 en Lecture

LireFichier 4, Truc

Nom ← Mid(Truc, 1, 20)

Prénom ← Mid(Truc, 21, 15)

Tel ← Mid(Truc, 36, 10)

Mail ← Mid(Truc, 46, 20)

L'instruction *LireFichier* récupère donc dans la variable spécifiée l'enregistrement suivant dans le fichier... "suivant", oui, mais par rapport à quoi ? Par rapport au dernier enregistrement lu. C'est en cela que le fichier est dit séquentiel. En l'occurrence, on récupère donc la première ligne, donc, le premier enregistrement du fichier, dans la variable Truc. Ensuite, le fichier étant organisé sous forme de champs de largeur fixe, il suffit de tronçonner cette variable Truc en autant de morceaux qu'il y a de champs dans l'enregistrement, et d'envoyer ces tronçons dans différentes variables. Et le tour est joué.

La suite du raisonnement s'impose avec une logique impitoyable : lire un fichier séquentiel de bout en bout suppose de programmer une *boucle*. Comme on sait rarement à l'avance combien d'enregistrements comporte le fichier, la combine consiste neuf fois sur dix à utiliser la fonction EOF (acronyme pour End Of File). Cette fonction renvoie la valeur Vrai si on a atteint la fin du fichier (auquel cas une lecture supplémentaire déclencherait une erreur).

L'algorithme, ultra classique, en pareil cas est donc :

Caractère : Truc

Ouvrir "Exemple.txt" sur 5 en Lecture

Tant Que Non EOF(5)

 LireFichier 5, Truc

 ...

Fin Tant Que

Fermer 5

Et neuf fois sur dix également, si l'on veut stocker au fur et à mesure en mémoire vive les informations lues dans le fichier, on a recours à un ou plusieurs *tableaux*. Et comme on ne sait pas d'avance combien il y aurait d'enregistrements dans le fichier, on ne sait pas davantage combien il doit y avoir d'emplacements dans les tableaux. Qu'importe, les programmeurs avertis que vous êtes connaissent la combine des tableaux dynamiques.

En rassemblant l'ensemble des connaissances acquises, nous pouvons donc écrire le prototype du code qui effectue la lecture intégrale d'un fichier séquentiel, tout en recopiant l'ensemble des informations en mémoire vive :

Tableaux Nom(), Prénom(), Tel(), Mail() en **Caractère**

Ouvrir "Exemple.txt" **sur 5 en Lecture**

$i \leftarrow -1$

Tant Que Non EOF(5)

LireFichier 5, Truc

$i \leftarrow i + 1$

Redim Nom(i)

Redim Prénom(i)

Redim Tel(i)

Redim Mail(i)

Nom(i) \leftarrow Mid(Truc, 1, 20)

Prénom(i) \leftarrow Mid(Truc, 21, 15)

Tel(i) \leftarrow Mid(Truc, 36, 10)

Mail(i) \leftarrow Mid(Truc, 46, 20)

Fin Tant Que

Fermer 5

Ici, on a fait le choix de recopier le fichier dans quatre tableaux distincts. On aurait pu également tout recopier dans un seul tableau : chaque case du tableau aurait alors été occupée par une ligne complète (un enregistrement) du fichier. Cette solution nous aurait fait gagner du temps au départ, mais elle alourdit ensuite le code, puisque chaque fois que l'on a besoin d'une information au sein d'une case du tableau, il faudra aller procéder à une extraction via la fonction MID. Ce qu'on gagne par un bout, on le perd donc par l'autre.

Mais surtout, comme on va le voir bientôt, il y a autre possibilité, bien meilleure, qui cumule les avantages sans avoir aucun des inconvénients.

Néanmoins, ne nous impatientons pas, chaque chose en son temps, et revenons pour le moment à la solution que nous avons employée ci-dessus.

Pour une opération d'écriture, ou d'ajout, il faut d'abord impérativement, sous peine de semer la panique dans la structure du fichier, constituer une chaîne équivalente à la nouvelle ligne du fichier. Cette chaîne doit donc être « calibrée » de la bonne manière, avec les différents champs qui « tombent » aux emplacements corrects. Le moyen le plus simple pour s'épargner de longs traitements est de procéder avec des chaînes correctement dimensionnées dès leur déclaration (la plupart des langages offrent cette possibilité) :

Ouvrir "Exemple.txt" **sur 3 en Ajout**

Caractère Truc

Caractère Nom*20, Prénom*15, Tel*10, Mail*20

Une telle déclaration assure que quel que soit le contenu de la variable Nom, par exemple, celle-ci comptera toujours 20 caractères. Si son contenu est plus petit, alors un nombre correct d'espaces sera automatiquement ajouté pour combler. Si on tente d'y entrer un contenu trop long, celui-ci sera automatiquement tronqué. Voyons la suite :

```
Nom ← "Jokers"  
Prénom ← "Midnight"  
Tel ← "0348946532"  
Mail ← "allstars@rockandroll.com"  
Truc ← Nom & Prénom & Tel & Mail  
EcrireFichier 3, Truc
```

Et pour finir, une fois qu'on en a terminé avec un fichier, il ne faut pas oublier de fermer ce fichier. On libère ainsi le canal qu'il occupait (et accessoirement, on pourra utiliser ce canal dans la suite du programme pour un autre fichier... ou pour le même).

Partie 02 : Exercices

Exercice 1 :

Ecrire un algorithme qui permet de :

Saisir deux entiers

Calculer la somme, différence, multiplication, et la division de ses deux entiers.

Afficher le résultat des quatre opérations

Exercice 2 :

Soit A, B, C, D quatre variables réel ; écrire un algorithme qui fait leur permutation cyclique donc le sens d'une aiguille de montre.

Exercice 3

Ecrire un algorithme qui permet de saisir deux entiers et afficher le plus grand d'entre eux.

Exercice 4

Ecrire un algorithme qui permet de saisir trois entiers et afficher le plus grand d'entre eux

Exercice 5

Ecrire l'algorithme qui permet de résoudre l'équation suivante :

$$ax+b=0$$

Exercice 6

Ecrire l'algorithme qui permet de résoudre l'équation suivante :

$$ax^2+bx+c=0$$

Exercice 7

Un patron décide de calculer le montant de participation au pris du repas de ses employés de la façon suivante :

- Si l'employé est célibataire la participation est de 20%
- S'il est marié la participation de 25%
- S'il a des enfants la participation est de 10% supplémentaire par enfants
- La participation est plafonné à 50%
- Si le salaire mensuelle est inférieur à 6000Dhs la participation est majoré de 10%

Ecrire l'algorithme qui lis les informations nécessaires au clavier et affiche pour un salarié la participation a la quel il a droit

Exercice 8

Soit un étudiant ayant 5 notes dans l'examen .

Ecrire un algorithme qui calcule la moyenne de cette étudiant sachant que tout les notes ont le même coefficient.

Exercice 9

Ecrire un algorithme qui calcule et affiche la table de multiplication d'un entier saisi par l'utilisateur.

Exercice 10

Ecrire un algorithme qui calcule et affiche le résultat de la suite

$$S = 1+2+\dots+n.$$

Exercice 11

Ecrire un algorithme qui affiche et calcule résultat de la suite :

$$S = 1-2+3-4+\dots(+/-)n.$$

Exercice 12

Ecrire l'algorithme qui permet de calculer et afficher le factoriel d'un entier N.
 $N! = N*(N-1)*(N-2)\dots*1.$

Exercice 13

Reprendre l'exercice du boultine de scolaire (exercice 8) sachant que le nombre de notes de l'étudiant n'est pas connu et les coefficients des matières sont différents.

Exercice 14

Ecrire un algorithme qui affiche le maximum, le minimum, la somme et la moyenne d'un nombre n entier .

n : n'est pas connu à l'avance .

(vous pouvez utiliser la technique de l'arrêt de la boucle on saisissons le chiffre 0).

Exercice 15

On désire calculer le montant de la facture d'électricité d'un abonner de la façon suivante :

Les frais fixe d'abonnement 70 Dhs

La consommation selon un tarif à tranche :

* 0.9 dh par kilowattheure pour les 110 premier kilowattheure.

* 0.98 dh par kilowattheure les 110 suivant.

* 1.20 dh par kilowattheure les kilowattheures supérieur à 220 kilowattheure.

Ecrire un algorithme qui permet de :

- Saisir le code de l'abonner
- Saisir le nom de l'abonner
- Saisir le totale de consommation d'électricité par mois.
- Calculer et afficher le montant hors taxe par tranche.
- Calculer et afficher le montant TTC par tranche.
- Calculer et afficher le montant totale de la facture
- Modifier votre programme pour qu'il puissent calculer la facture de plusieurs clients.

Exercice 16

Ecrire un algorithme qui permet de :

- Saisir le nom et le prénom de l'utilisateur
- Saisir le jour, le mois, l'année de naissance de l'utilisateur.
- Saisir le jour, le mois, l'année actuel

Un contrôle de validité des dates saisis :

- Les jours doivent être compris entre 1 et 31
- Les mois doivent être compris entre 1 et 12
- L'année doit être obligatoirement inferieure ou égale à 2010

Si une date n'est pas valide un message d'erreur s'affiche à l'utilisateur et lui demande s'il veut recommencez.

Si les valeurs sont valide ,calculer l'âge en nombre d'année entière , et afficher à l'utilisateur le message :

« Bonjour M(+nom+prénom+) vous avez (âge) ans »

Si le jour et le mois de naissance correspondent au jour et mois actuel afficher à l'utilisateur un message « Joyeux anniversaire »

Exercice 17

Ecrire un algorithme qui permet de lire une suite n entier dans un tableau, lire un nombre entier comme donnée et chercher si se nombre existe dans le tableau ; et afficher en plus la position de l'entier s'il existe dans le tableau.

Exercice 18

On saisi des entiers et on les ranges dans un tableau(maximum 50 entiers).
Ecrire un algorithme qui affiche le maximum, le minimum et la valeur moyenne de ses entiers. Le programme doit permettre de contrôler la saisi(si l'utilisateur souhaite saisir moins de 50 entiers il saisi le chiffre 0) ; et afficher en plus la position du maximum , et la position du minimum dan le tableau.

Exercice 19

Ecrire un algorithme qui permet de saisir une liste de classe comprenant le nom de chaque stagiaire et sa moyenne générale, et affiche les noms et les moyennes de tous les stagiaire qui on une moyenne supérieure a la moyenne de la classe.
Le programme doit aussi afficher le nom et la moyenne du premier et du dernier de la classe.

Exercice 20

Un wagon comporte 60 places assise dont 30 places pour les fumeurs numéroté de 1 à 30, et les non-fumeurs numéroté de 31 à 60.
Ecrire un algorithme permettant de faire la réservation de places des wagons et d'arrêter s'il n y a aucune place libre .
On voudrez avoir la possibilité d'annuler une réservation faite par un client (sachant que ce dernier peut oublier ou égarer le numéro de sa place) , et de réserver les places annuler pour d'autre personne.

Exercice 21

Le tri par sélection :

Le premier algorithme auquel on pense pour effectuer ce tri est celui-ci :
On cherche le plus petit élément du tableau et on le place en 1er, puis on cherche le plus petit dans ce qui reste et on le met en second, etc.

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 52 | 10 | 1 | 25 | 62 | 3 | 8 | 55 | 3 | 23 |
| 1 | 52 | 10 | 25 | 62 | 3 | 8 | 55 | 3 | 23 |
| 1 | 3 | 52 | 10 | 25 | 62 | 8 | 55 | 3 | 23 |
| 1 | 3 | 3 | 52 | 10 | 25 | 62 | 8 | 55 | 23 |
| 1 | 3 | 3 | 8 | 52 | 10 | 25 | 62 | 55 | 23 |
| 1 | 3 | 3 | 8 | 10 | 52 | 25 | 62 | 55 | 23 |
| 1 | 3 | 3 | 8 | 10 | 23 | 52 | 25 | 62 | 55 |
| 1 | 3 | 3 | 8 | 10 | 23 | 25 | 52 | 62 | 55 |
| 1 | 3 | 3 | 8 | 10 | 23 | 25 | 52 | 62 | 55 |
| 1 | 3 | 3 | 8 | 10 | 23 | 25 | 52 | 55 | 62 |

Ecrire l'algorithme qui permet de réaliser ce tri.

Exercice 22

Le tri par bulle :

Un tri bulle est un tri plus astucieux, son principe est de faire remonter petit à petit un élément trop grand vers le haut du tableau en comparant les éléments deux à deux. Si l'élément de gauche est supérieur à son voisin de droit on les inverse et on continue avec le suivant. Lorsque l'on est en haut du tableau on repart au début et on s'arrête lorsque tous les éléments sont bien placés.

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 52 | 10 | 1 | 25 | 62 | 3 | 8 | 55 | 3 | 23 |
| 10 | 52 | 1 | 25 | 62 | 3 | 8 | 55 | 3 | 23 |
| 10 | 1 | 52 | 25 | 62 | 3 | 8 | 55 | 3 | 23 |
| 10 | 1 | 52 | 25 | 62 | 3 | 8 | 55 | 3 | 23 |
| 10 | 1 | 25 | 52 | 62 | 3 | 8 | 55 | 3 | 23 |
| 10 | 1 | 25 | 52 | 3 | 62 | 8 | 55 | 3 | 23 |
| 10 | 1 | 25 | 52 | 3 | 8 | 62 | 55 | 3 | 23 |
| 10 | 1 | 25 | 52 | 3 | 8 | 55 | 62 | 3 | 23 |
| 10 | 1 | 25 | 52 | 3 | 8 | 55 | 3 | 62 | 23 |
| 10 | 1 | 25 | 52 | 3 | 8 | 55 | 3 | 23 | 62 |

On a parcouru tout le tableau, on recommence, jusqu'à ce que tout soit bien placé.
Ecrire l'algorithme qui réalise ce tri.

Exercice 23

Le tri par permutation :

Le tri par permutation est le tri du jeu de cartes.

On parcourt le tableau jusqu'à ce que l'on trouve un élément plus petit que le précédent, donc mal placé. On prend cet élément et on le range à sa place dans le tableau puis on continue la lecture. On s'arrête à la fin du tableau.

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 52 | 10 | 1 | 25 | 62 | 3 | 8 | 55 | 3 | 23 |
| 10 | 52 | 1 | 25 | 62 | 3 | 8 | 55 | 3 | 23 |
| 1 | 10 | 52 | 25 | 62 | 3 | 8 | 55 | 3 | 23 |
| 1 | 10 | 25 | 52 | 62 | 3 | 8 | 55 | 3 | 23 |
| 1 | 3 | 10 | 25 | 52 | 62 | 8 | 55 | 3 | 23 |
| 1 | 3 | 8 | 10 | 25 | 52 | 62 | 55 | 3 | 23 |
| 1 | 3 | 8 | 10 | 25 | 52 | 62 | 55 | 3 | 23 |
| 1 | 3 | 8 | 10 | 25 | 52 | 55 | 62 | 3 | 23 |
| 1 | 3 | 3 | 8 | 10 | 25 | 52 | 55 | 62 | 23 |
| 1 | 3 | 3 | 8 | 10 | 23 | 25 | 52 | 55 | 62 |

Ecrire l'algorithme qui réalise ce tri.

Exercice 24

Le tri par comptage :

Le tri par comptage consiste pour chaque élément du tableau à compter combien d'éléments sont plus petits que lui, grâce à ce chiffre on connaît sa position dans le tableau résultat.

| | | | | | | | | | | |
|-------|----|----|---|----|----|---|---|----|---|----|
| | 52 | 10 | 1 | 25 | 62 | 3 | 8 | 55 | 3 | 23 |
| Nbr : | 7 | 4 | 0 | 6 | 9 | 1 | 3 | 8 | 1 | 5 |
| Pos : | 8 | 5 | 1 | 7 | 10 | 2 | 4 | 9 | 3 | 6 |

Exercice 25

Ecrire un programme qui permet de faire des étoiles avec la forme suivante :

```
*****
*****
*****
*****
*****
*****
****
***
**
*
```

Exercice 26

Ecrire un programme qui permet de faire des étoiles avec la forme suivante :

```
      *
     **
    ***
   ****
  *****
 *****
*****
*****
*****
*****
```

Exercice 27

Ecrire un programme qui permet de faire des étoiles avec la forme suivante :

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

Exercice 28

Ecrire un programme qui permet de faire des étoiles avec la forme suivante :

```
      *
     ***
    *****
   *****
  *****
 *****
*****
*****
```


Exercice 33

Pour un entier n strictement positif on associe $n/2$ si n est paire et $3n+1$ si n est impaire. En réappliquant cette transformation à l'entier obtenu, on définit un algorithme dit de Syracuse. On admettra que pour tout entier strictement positif de départ on finisse toujours par arriver à 1. On demande d'écrire un programme qui, pour une valeur de départ proposée par l'utilisateur, **affiche la liste des entiers obtenus jusqu'à 1, ainsi que le nombre de fois qu'il est nécessaire d'appliquer la transformation pour y arriver.**

Voici un exemple de déroulement de cet algorithme :

Valeur de départ (entier strictement positif) ? **12**

6 3 10 5 16 8 4 2 1

On doit appliquer **9** fois la transformation avant d'arriver à **1**

Exercice 34

Ecrire un programme qui transfère une matrice M à deux dimensions L et C dans un tableau V à une dimension .

Exercice 35

La direction d'une entreprise désire automatiser le calcul de l'indemnité à verser aux cadres en cas de licenciement.

Après un an d'ancienneté dans l'entreprise, il sera alloué aux cadres licenciés une indemnité tenant compte de leur ancienneté et s'établissant comme suit :

- la moitié du salaire d'un mois par année d'ancienneté : pour la tranche d'ancienneté entre 1 et 10 ans
- au delà de 10 ans un mois de salaire par année d'ancienneté
- une indemnité supplémentaire serait alloué aux cadres âgés de plus de 45 ans de :
 - 2 mois si le cadre est âgé de 46 à 49 ans
 - 5 mois si le cadre est âgé de plus de 50 ans.

Ecrire un programme qui permet de saisir l'âge, l'ancienneté et le dernier salaire et d'afficher l'indemnité du cadre.

Exercice 36

Ecrire un algorithme qui effectue la lecture d'une matrice carrée A ainsi que sa taille n et affiche la matrice transposée tA de A (Pour une matrice $A(i,j)$, $tA(j,i)$).

Exercice 37

Ecrire un algorithme qui effectuer la lecture d'une matrice carrée A ainsi que sa taille n et affiche la trace de A .

(Pour une matrice $A(a_{i,j})$, $\text{Trace}(A) = \sum a_{i,i}$ la somme des éléments sur la diagonale).

Exercice 38

Ecrire un programme qui permet d'insérer une valeur X dans un tableau T , supposé trié, de façon à respecter l'ordre des éléments de T . le tableau T contient N éléments et sera dimensionné à $N+1$ (pour permettre de ranger X)

Si $N=10$ et $T=$

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 17 | 17 | 21 | 23 | 24 | 26 | 27 | 30 | 30 | 38 |
|----|----|----|----|----|----|----|----|----|----|

Si $X=25$ on doit obtenir : $T =$

| | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|
| 17 | 17 | 21 | 23 | 24 | 25 | 26 | 27 | 30 | 30 | 38 |
|----|----|----|----|----|----|----|----|----|----|----|

Exercice39

Ecrire un programme qui lit un entier X et un tableau A du type **int** au clavier et élimine toutes les occurrences de X dans A en tassant les éléments restants (décalage).

Exercice 40

Faire un programme pour le calcul et l'affichage suivant :

```
1 * 8 + 1 = 9
12 * 8 + 2 = 98
123 * 8 + 3 = 987
1234 * 8 + 4 = 9876
12345 * 8 + 5 = 98765
123456 * 8 + 6 = 987654
1234567 * 8 + 7 = 9876543
12345678 * 8 + 8 = 98765432
123456789 * 8 + 9 = 987654321
```

Exercice 41

Ecrire un programme qui saisit une chaîne pouvant contenir des espaces et qui affiche chaque mot de la chaîne, le séparateur étant l'espace.

Exemple, on tape : je pense donc je suis

Le programme affiche :

```
Mot 1 : je
Mot 2 : pense
Mot 3 : donc
Mot 4 : je
Mot 5 : suis
```

Exercice 42

Ecrire une fonction f ayant en paramètres un tableau t1 de taille quelconque et un entier n indiquant la taille du tableau, ainsi qu'un tableau t2 de la même taille que t1. f doit renvoyer par un return un entier nb indiquant le nombre de valeurs comprises entre 0 et 10 dans le tableau t1. f doit mettre dans le tableau t2 les différentes valeurs comprise entre 0 et 10 qu'il a rencontrées dans le tableau t1.

Exercice 43

Ecrire un programme de recherche de la valeur maximale et minimale d'un **tableau [N][M]** de réels de taille $N \times M$.

Exercice 44

Faire un programme permettant de calculer d'afficher la table des produits pour N variant de 1 à 10 :

Programmation structurée

| X*Y | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----|---|----|----|----|----|----|----|----|----|----|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 3 | 0 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 4 | 0 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 |
| 5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
| 6 | 0 | 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60 |
| 7 | 0 | 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70 |
| 8 | 0 | 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80 |
| 9 | 0 | 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90 |
| 10 | 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |

Exercice 45

Faire un programme qui calcule le produit scalaire de deux vecteurs d'entiers U et V (de même dimension).

Exemple

$$\begin{matrix} / & & \backslash & / & & \backslash \\ | & 3 & 2 & -4 & | & * & | & 2 & -3 & 5 & | \\ \backslash & & / & & \backslash & & / & & \backslash & & / \end{matrix} = 3*2 + 2 * (-3) + (-4)*5 = -20$$

Exercice 46

On dispose de deux tableaux A et B (de dimensions respectives N et M), triés par ordre croissant. Fusionner les éléments de A et B dans un troisième tableau FUS trié par ordre croissant.

Exercice 47

Faire un programme qui construit le triangle de PASCAL de degré N et le mémorise dans une matrice carrée P de dimension N+1.

Exemple : *Triangle de Pascal de degré 6* :

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
    
```

Exercice 48

Ecrire un programme qui transfère une matrice M à deux dimensions L et C dans un tableau V à une dimension .

Exemple :

$$\begin{array}{c}
 / \quad \quad \backslash \\
 | \mathbf{a} \ \mathbf{b} \ \mathbf{c} \ \mathbf{d} | \quad \quad / \quad \quad \quad \backslash \\
 | \mathbf{e} \ \mathbf{f} \ \mathbf{g} \ \mathbf{h} | \quad \Rightarrow \quad | \mathbf{a} \ \mathbf{b} \ \mathbf{c} \ \mathbf{d} \ \mathbf{e} \ \mathbf{f} \ \mathbf{g} \ \mathbf{h} \ \mathbf{i} \ \mathbf{j} \ \mathbf{k} \ \mathbf{l} | \\
 | \mathbf{i} \ \mathbf{j} \ \mathbf{k} \ \mathbf{l} | \quad \quad \quad \backslash \quad \quad \quad / \\
 \backslash \quad \quad /
 \end{array}$$

Exercice 49

Ecrire un programme qui effectue la transposition tA d'une matrice A de dimensions N et M en un matrice de dimensions M et N .

Rappel :

$$\begin{array}{c}
 / \quad \quad \backslash \quad \quad / \quad \quad \backslash \\
 \mathbf{tA} = \mathbf{t} \quad | \mathbf{a} \ \mathbf{b} \ \mathbf{c} \ \mathbf{d} | \quad = \quad | \mathbf{a} \ \mathbf{e} \ \mathbf{i} | \\
 \quad \quad \quad | \mathbf{e} \ \mathbf{f} \ \mathbf{g} \ \mathbf{h} | \quad \quad \quad | \mathbf{b} \ \mathbf{f} \ \mathbf{j} | \\
 \quad \quad \quad | \mathbf{i} \ \mathbf{j} \ \mathbf{k} \ \mathbf{l} | \quad \quad \quad | \mathbf{c} \ \mathbf{g} \ \mathbf{k} | \\
 \quad \quad \quad \backslash \quad \quad / \quad \quad \quad | \mathbf{d} \ \mathbf{h} \ \mathbf{l} | \\
 \quad \backslash \quad \quad /
 \end{array}$$

Exercice 50

Ecrire un programme qui réalise l'addition de deux matrice A et B même dimensions N et M .

Rappel :

$$\begin{array}{c}
 / \quad \quad \backslash \quad \quad / \quad \quad \backslash \quad \quad / \quad \quad \quad \backslash \\
 | \mathbf{a} \ \mathbf{b} \ \mathbf{c} \ \mathbf{d} | \quad \quad | \mathbf{a}' \ \mathbf{b}' \ \mathbf{c}' \ \mathbf{d}' | \quad \quad | \mathbf{a+a}' \ \mathbf{b+b}' \ \mathbf{c+c}' \ \mathbf{d+d}' | \\
 | \mathbf{e} \ \mathbf{f} \ \mathbf{g} \ \mathbf{h} | \quad + \quad | \mathbf{e}' \ \mathbf{f}' \ \mathbf{g}' \ \mathbf{h}' | \quad = \quad | \mathbf{e+e}' \ \mathbf{f+f}' \ \mathbf{g+g}' \ \mathbf{h+h}' | \\
 | \mathbf{i} \ \mathbf{j} \ \mathbf{k} \ \mathbf{l} | \quad \quad | \mathbf{i}' \ \mathbf{j}' \ \mathbf{k}' \ \mathbf{l}' | \quad \quad | \mathbf{i+i}' \ \mathbf{j+j}' \ \mathbf{k+k}' \ \mathbf{l+l}' | \\
 \backslash \quad \quad / \quad \quad \quad \backslash \quad \quad / \quad \quad \quad \backslash \quad \quad /
 \end{array}$$

Exercice 51

Ecrire un algorithme qui calcule le **schtroumpf** des deux tableaux. Pour calculer le schtroumpf, il faut multiplier chaque élément du tableau 1 par chaque élément du tableau 2, et additionner le tout.

Exercice 52

Ecrire un programme qui demande à l'utilisateur de taper 10 entiers compris entre 0 et 20 qui seront stockés dans un tableau et qui affiche le nombre de fois qu'on a tapé un 0, le nombre de 1, le nombre de 2, ..., le nombre de 20.

Exercice 53

Considérant un tableau numérique «**T**» de **N** éléments, et un deuxième tableau numérique «**V**» de **M** éléments, concevoir un programme qui permet de poser les éléments des deux tableaux dans un troisième tableau numérique «**R**» de telle façon à j' avoir trié en ordre croissant.

Exercice 54

En utilisant les procédures et les fonctions , faire un menu qui permet de gérer les tableaux , (des tableaux triée avec différentes tri, chercher le minimum ,le maximum dans un tableaux.ect.)

Exemple :

Menu de Gestion des tableaux

- 1- Remplir le tableau**
- 2- Chercher un entier dans un tableau**
- 3- Chercher le minimum**
- 4- Chercher le maximum**
- 5-Triée le tableau par sélection**
- 6- Triée le tableau par bulle**
- 7- Triée le tableau par permutation**
- 8- Triée le tableau par comptage**
- 9-Quitter**

Exercice 55

En utilisant une / ou plusieurs structures, faite un menu de la gestion d'une activité de votre propre choix

Exemple :

Menu du la gestion de l'activité

- 1) Afficher les informations de l'activité**
- 2) Chercher une information**
- 3) Modifier une information**
- 4) Supprimer une information**

5)

n) Quitter le menu

Exercice 56

Ecrire un programme C++ qui reçoit deux tableaux de même taille n triés dans l'ordre croissant, puis chercher et afficher leur premier élément commun.

Exercice 57 cryptographie 1

Un des plus anciens systèmes de cryptographie (aisément déchiffrable) consiste à décaler les lettres d'un message pour le rendre illisible. Ainsi, les A deviennent des B, les B des C, etc. Ecrivez un algorithme qui demande une phrase à l'utilisateur et qui la code selon ce principe. Comme dans le cas précédent, le codage doit s'effectuer au niveau de la variable stockant la phrase, et pas seulement à l'écran.

Exercice 58 cryptographie 2- le chiffre de César

Une amélioration (relative) du principe précédent consiste à opérer avec un décalage non de 1, mais d'un nombre quelconque de lettres. Ainsi, par exemple, si l'on choisit un décalage de 12, les A deviennent des M, les B des N, etc.

Réalisez un algorithme sur le même principe que le précédent, mais qui demande en plus quel est le décalage à utiliser. Votre sens proverbial de l'élégance vous interdira bien sûr une série se vingt-six 'Si ...Alors'

Exercice 59 cryptographie 3

Une technique ultérieure de cryptographie consista à opérer non avec un décalage systématique, mais par une substitution aléatoire. Pour cela, on utilise un alphabet-clé, dans lequel les lettres se succèdent de manière désordonnée, par exemple :

H Y L U J P V R E A K B N D O F S Q Z C W M G I T X

C'est cette clé qui va servir ensuite à coder le message. Selon notre exemple, les A deviendront des H, les B des Y, les C des L, etc.

Ecrire un algorithme qui effectue ce cryptage (l'alphabet-clé sera saisi par l'utilisateur, et on suppose qu'il effectue une saisie correcte).

Exercice 60 Contraction

Recopier une phrase dans une autre en ôtant toutes les occurrences d'un caractère

Soit une phrase terminée par un point.

Il s'agit de la restituer en supprimant les occurrences d'un caractère donné.

Exemple

Phrase : abbcccdeeeffg

Caractère : c

Résultat : abbdeeeffg

Donnez le jeu d'essai qui permet de tester cette procédure.

Donnez l'algorithme de la procédure en pseudo code.

Exercice 61 Doublons

Recopier une phrase dans une autre en ôtant tous les doublons de caractères successifs

Soit une phrase terminée par un point.

Il s'agit de la restituer en supprimant tous les doublons de caractères successifs.

Exemple : abbcccddeeffg. donne
 abcdefg.

Donnez le jeu d'essai qui permet de tester cette procédure.

Pour tester le programme, c'est-à-dire voir s'il répond bien à nos attentes, s'il n'a pas de « bug », avant de la faire « tourner » sur la machine nous imaginons un jeu d'essai avec les cas à tester et le résultat attendu pour chaque cas : **c'est le jeu d'essai** .

Donnez l'algorithme de la procédure.

Exercice 62 Equivalence

Déterminer si deux phrases sont équivalentes.

Soit deux phrases terminées par un même terminateur.

Elles sont dites équivalentes si elles ont les mêmes lettres dans le même ordre mais avec un nombre d'occurrences de ces lettres qui peut différer entre les deux phrases.

On supposera qu'il existe une fonction longueur **lg** de chaîne qui renvoie un entier

Exemple :

 abbcccddeeffg
 abcdeffffg sont équivalentes

Donnez le jeu d'essai qui permet de tester cette procédure.

Donnez l'algorithme de la procédure toujours en pseudo code.

Exercice 63 Eparpillement

Chercher les lettres d'un mot éparpillement dans une phrase , dans le même ordre.

Soient un caractère terminateur et une phrase et une phrase terminée par ce caractère terminateur .

Soient un mot donné

Il s'agit de vérifier si les lettres du mot sont bien présentes dans la phrase, de dans le même ordre que celui du mot.

Exemple :

 terminateur : .
 phrase : le chat est gris et boit.
 mot : lattis
 longueur : 6

 donne vrai

Donnez l'algorithme de la procédure en pseudo code.

Exercice 64 Palindrome

Déterminer si une chaîne de caractères est palindrome.

Un palindrome est une phrase qui peut se lire dans les deux sens.

Les espaces sont ignorés.

Exemple : esope reste ici et se repose.

Le terminateur est ici un point.

Donnez l'algorithme du programme.

Exercice 65

- a- Faire l'algorithme pour calculer $K=1 * 1/2 * 1/3 * \dots * 1/n (n>0)$.
b- Ecrire le programme C++ permettant de calculer K. N'utiliser pas la classe Clavier.

Exercice 66

La direction d'un supermarché a décidé d'accorder des réductions à ses clients selon le montant d'achat

La réduction est calculée selon les règles suivantes :

- 20% pour un montant d'achat de plus de 5000 dhs
- 15% pour un montant d'achat entre 3000 dhs < MonantAchat ≤ 5000 dhs
- 10% pour un montant d'achat entre 1000 dhs < MonantAchat ≤ 3000 dhs
- Aucune réduction pour un montant d'achat inférieur à 1000 dhs

Ecrire un programme qui permet de calculer et d'afficher la réduction et montant à payer.

Exercice 67

Ecrire un programme permettant de saisir le prix unitaire et la quantité commandée d'un article.

Le programme affichera le prix à payer , le port , et la remise sachant que :

- Le porte est gratuit si le montant hors taxe est supérieur à 1000 dh
- Le porte est 3% dans le cas contraire
- La remise est de 5% si le montant hors taxe est compris entre 300 et 1000 et de 10% au-delà de 1000

NB : Montant hors Taxe = prix unitaire × quantité commandée

Exercice 68

Ecrire un programme qui demande un entier N positif en base 10, et un entier Ba (égale à 2, 8 ou 16) et convertit N en base Ba.

Exercice 69

Ecrire un programme de faire le tri dans l'ordre croissant et décroissant d'une matrice de taille N × M

Exercice 70

Que produite l'algorithme suivant ?

Tableau Suite(7) en Entier

Variable i en Entier

Début

Suite (0) ← 1

Suite (1) ← 1

Pour i ← 2 à 7

Suite (i) ← Suite(i-1) + Suite(i-2)

i suivant

Pour i ← 0 à 7

Ecrire Suite(i)
i suivant

Fin

Exercice 71

Soit une classe de 20 stagiaires. Chaque stagiaire est représenté par les informations suivantes :

| | |
|------------|--------------|
| Nom | chaîne |
| Prénom | chaîne |
| Notes | Tableau réel |
| Moyenne | réel |
| Classement | entier |

Utilisez un tableau pour contenir les données des stagiaires.

On veut réaliser les traitements suivants :

- Saisir les données nécessaires
- Calculer la moyenne pour chaque stagiaire
- Trier les stagiaires par la moyenne et dans le sens décroissant.
- Déterminer le classement pour chaque stagiaire.
- Afficher les données de tous les stagiaires.

Exercice 72

On considère une séquence d'entiers s de longueur L représentée dans un tableau T d'entiers défini sur l'intervalle $[1 \dots L_{\max}]$, $0 < L < L_{\max}$.

On veut écrire un programme qui remplace dans T la suite s par la suite s' de longueur L' (avec $L' \leq L$), déduite de s en supprimant tous les éléments redondants. Un élément de s est redondant s'il est égal à un autre élément de s .

Le programme ne doit pas utiliser de tableau intermédiaire pour créer s' . L'ordre des éléments reste celui de la séquence de départ. Etudier tout d'abord le problème en supposant que T peut ne pas être trié en ordre croissant.

Exemple :

Si $s = [15,4,19,4,8,11,11,3,4,19]$ et $L = 10$ alors $s' = [15,4,19,8,11,3]$ et $L = 6$

Exercice 73

Ecrire un programme qui demande à l'utilisateur de taper 10 entiers qui seront stockés dans un tableau. Le programme doit ensuite afficher soit "le tableau est croissant", soit "le tableau est décroissant", soit "le tableau est constant", soit "le tableau est quelconque".

Exercice 74

Ecrire une fonction qui a en paramètre une chaîne de caractères (paramètre en entrée et en sortie) et qui transforme toutes les minuscules de la chaîne en majuscules.

Exercice 75

Ecrire une fonction qui a en paramètre une chaîne de caractères (paramètre en entrée et en

sortie) et qui supprime toutes les voyelles

Exercice 76

Ecrire un programme qui demande à l'utilisateur de taper un verbe du premier groupe et qui le conjugue à l'indicatif présent.

Exercice 77

On veut gérer des produits dans un entrepôt. Un produit est défini par 2 chaînes de caractères : le code produit ("H567" par exemple et qui comporte au maximum 9 caractères utiles), l'intitulé ("pots de peinture" et qui comporte au maximum 99 caractères utiles) et un entier qui indique la quantité en stock (803 par exemple). Il faut gérer une liste de produits en veillant à ce qu'il n'y ait pas deux produits avec le même code produit. Bien sûr la quantité en stock ne peut pas être négative. La liste comportera au maximum 100 produits.

Il faut gérer la liste grâce au menu suivant :

1. Ajouter un produit (on tape le code produit et l'intitulé, la quantité est initialisée à 0).
2. Afficher la liste de produits.
3. Supprimer un produit en tapant le code produit.
4. Acheter un produit en tapant le code produit et la quantité achetée.
5. Vendre un produit en tapant le code produit et la quantité vendue.
6. Quitter

On veillera à bien décomposer ce problème en différents modules et à mener une réflexion sur les fonctions nécessaires dans chaque module.

Exercice 78

En utilisant une structure point définissant un abscisse et un ordonné, écrire un programme qui permet de :

- saisir les coordonnées d'un point au clavier.
- afficher les coordonnées d'un point.
- calculer la distance entre 2 points.

calculer le milieu de 2 points

Exercice 79

On appelle nombre d'Armstrong un nombre qui est égale à la somme des cubes de ses chiffres. on se propose de construire un algorithme qui recherche les nombres d'ARMSTRONG entre 0 et un entier saisi au clavier

Exemple : $371 = 3^3 + 7^3 + 1^3$

Exercice 80

Ecrire un algorithme qui permet de définir si un nombre est Armstrong ou non.

Partie 03 : Correction des exercices

1. INTRODUCTION

1.1. LES LANGAGES INFORMATIQUE

On appelle « *langage informatique* » un langage destiné à décrire l'ensemble des actions consécutives qu'un ordinateur doit exécuter. Un langage informatique est ainsi une façon pratique pour nous (humains) de donner des instructions à un ordinateur.

A contrario, le terme « langage naturel » représente les possibilités d'expression partagé par un groupe d'individus (par exemple l'anglais ou le français).

Les langages servant aux ordinateurs à communiquer entre eux n'ont rien à voir avec des langages informatiques, on parle dans ce cas de protocoles de communication, ce sont deux notions totalement différentes. Un langage informatique est rigoureux :

À CHAQUE instruction correspond UNE action du processeur.

Le langage utilisé par le processeur est appelé *langage machine*. Il s'agit des données telles qu'elles arrivent au processeur, constituées d'une suite de 0 et de 1 (données binaire).

Le langage machine n'est ainsi pas compréhensible par l'être humain, c'est pourquoi des langages intermédiaires, compréhensibles par l'homme, ont été mis au point. Le code écrit dans ce type de langage est transformé en langage machine pour être exploitable par le processeur.

L'assembleur est le premier langage informatique qui ait été utilisé. Celui-ci est très proche du langage machine mais reste compréhensible pour des développeurs. Toutefois, un tel langage est tellement proche du langage machine qu'il dépend étroitement du type de processeur utilisé (chaque type de processeur peut avoir son propre langage machine). Ainsi, un programme développé pour une machine ne pourra pas être *porté* sur un autre type de machine. Le terme « *portabilité* » désigne l'aptitude d'un programme informatique à être utilisé sur des machines de types différents. Pour pouvoir utiliser un programme informatique écrit en assembleur sur un autre type de machine, il sera parfois nécessaire de réécrire entièrement le programme !

Un langage informatique a donc plusieurs avantages :

- ∂ il est plus facilement compréhensible que le langage machine ;
- ∂ il permet une plus grande portabilité, c'est-à-dire une plus grande facilité d'adaptation sur des machines de types différents ;

Langages impératifs et fonctionnels

On distingue habituellement deux grandes familles de langages de programmation, selon la manière de laquelle les instructions sont traitées :

- ❖ les langages impératifs ;
- ❖ les langages fonctionnels.

Langage impératif

Un langage impératif organise le programme sous forme d'une série d'instructions, regroupées par blocs et comprenant des sauts conditionnels permettant de revenir à un bloc d'instructions si la condition est réalisée. Il s'agit historiquement des premiers langages, même si de nombreux langages modernes utilisent toujours ce principe de fonctionnement.

Les langages impératifs structurés souffrent néanmoins d'un manque de souplesse étant donné le caractère séquentiel des instructions.

Langage fonctionnel

Un *langage fonctionnel* (parfois appelé *langage procédural*) est un langage dans lequel le programme est construit par fonctions, retournant un nouvel état en sortie et prenant en entrée

la sortie d'autres fonctions. Lorsque la fonction s'appelle elle-même, on parle alors de récursivité.

Interprétation et compilation

Les langages informatiques peuvent grossièrement se classer en deux catégories :

∂ les langages interprétés

∂ les langages compilés.

Langage interprété

Un langage informatique est par définition différent du langage machine. Il faut donc le traduire pour le rendre intelligible du point de vue du processeur. Un programme écrit dans un langage interprété a besoin d'un programme auxiliaire (l'interpréteur) pour traduire au fur et à mesure les instructions du programme.

Langage compilé

Un programme écrit dans un langage dit « *compilé* » va être traduit une fois pour toutes par un programme annexe, appelé *compilateur*, afin de générer un nouveau fichier qui sera autonome, c'est-à-dire qui n'aura plus besoin d'un programme autre que lui pour s'exécuter; on dit d'ailleurs que ce fichier est *exécutable*.

Un programme écrit dans un langage compilé a comme avantage de ne plus avoir besoin, une fois compilé, de programme annexe pour s'exécuter. De plus, la traduction étant faite une fois pour toute, il est plus rapide à l'exécution.

Toutefois il est moins souple qu'un programme écrit avec un langage interprété car à chaque modification du fichier source (fichier intelligible par l'homme: celui qui va être compilé) il faudra recompiler le programme pour que les modifications prennent effet.

D'autre part, un programme compilé a pour avantage de garantir la sécurité du code source. En effet, un langage interprété, étant directement intelligible (lisible), permet à n'importe qui de connaître les secrets de fabrication d'un programme et donc de copier le code voire de le modifier. Il y a donc risque de non-respect des droits d'auteur. D'autre part, certaines applications sécurisées nécessitent la confidentialité du code pour éviter le piratage (transaction bancaire, paiement en ligne, communications sécurisées, ...).

Langages intermédiaires

Certains langages appartiennent en quelque sorte aux deux catégories (LISP, Java, Python, ..) car le programme écrit avec ces langages peut dans certaines conditions subir une phase de compilation intermédiaire vers un fichier écrit dans un langage qui n'est pas intelligible (donc différent du fichier source) et non exécutable (nécessité d'un interpréteur). Les applets Java, petits programmes insérés parfois dans les pages Web, sont des fichiers qui sont compilés mais que l'on ne peut exécuter qu'à partir d'un navigateur internet (ce sont des fichiers dont l'extension est .class).

1.2. Correction en langage C++

On peut considérer que C++ « est du C » avec un ajout de fonctionnalités. Une remarque importante est à faire cependant : certains programmes syntaxiquement corrects en C ne le sont pas en C++.

Programmation structurée

| | |
|--|--|
| <p>ligne]</p> <ul style="list-style-type: none"> • <u>afficher</u> « La somme est », D ; [et revenir à la ligne] • <u>afficher</u> «Donner... »,D ,«cogner... » ; | <ul style="list-style-type: none"> • cout << " La somme est " << D << endl ; (l'affichage de la valeur de D est sur la meme ligne que la phrase) • cout<< " Donner... " <<D<< " cogner... " ; |
| <ul style="list-style-type: none"> • <u>saisir</u> C ; | <ul style="list-style-type: none"> • cin >> C ; |
| <ul style="list-style-type: none"> • afficher aux coordonnées indiqués : <u>Cadrer</u> (x,y) ; <u>Afficher</u> | <ul style="list-style-type: none"> • goto(1,2) ; instructions ; |
| <ul style="list-style-type: none"> • <u>Tout Effacer Ecran</u> ; | <ul style="list-style-type: none"> • clrscr () ; |
| <ul style="list-style-type: none"> • <u>Pause</u> ; | <ul style="list-style-type: none"> • getch() ; |
| <ul style="list-style-type: none"> • <u>Stopper</u> ; | <ul style="list-style-type: none"> • break ; |
| <ul style="list-style-type: none"> • <u>Si</u> (condition) <u>alors</u> Instruction 1 ; Instruction n ; <u>fsi</u> ; (; non obligatoire après <u>fsi</u>) • <u>Si</u> (condition) <u>alors</u> Instruction 1 ; Instruction n ; <u>Sinon</u> Instruction 1 ; Instruction n ; <u>fsi</u> ; • <u>Si</u> (Qt>0 <u>et</u> Qt<30) <u>alors</u> | <ul style="list-style-type: none"> • if (condition) { instruction 1 ; instruction n ; } • if (condition) { instruction 1 ; instruction n ; } else { instruction 1 ; instruction n ; } • if ((Qt>0) && (Qt<30)) { |
| <ul style="list-style-type: none"> • = • ≠ • < • > • ≤ • ≥ | <ul style="list-style-type: none"> • == • != • < • > • <= • >= |
| <ul style="list-style-type: none"> • <u>Pour</u> i ← 1 <u>jusqu'à</u> 10 <u>faire</u> Instruction 1 ; | <ul style="list-style-type: none"> • for (i=1 ; i<=10 ; i++) [ici i++ veut dire i=i+1] { Instruction 1 ; |

Programmation structurée

| | |
|--|--|
| <p>Instruction n ;</p> <p><u>F</u>pour ;</p> <ul style="list-style-type: none"> • <u>R</u>épéter Instruction 1 ; <p>Instruction n ;</p> <p>i ← i + 1 ;</p> <p><u>t</u>ant que (i <=10) ;</p> <ul style="list-style-type: none"> • <u>T</u>ant que (i <10) <u>f</u>aire Instruction 1 ; <p>Instruction n ;</p> <p>i ← i + 1 ;</p> <p><u>f</u>tant que ;</p> | <p>Instruction n ;</p> <p>}</p> <ul style="list-style-type: none"> • do { instruction 1 ; <p>instruction n ;</p> <p>i=i+1 ;</p> <p>} while (i<=10) ;</p> <ul style="list-style-type: none"> • while (i<10) { instruction 1 ; <p>instruction n ;</p> <p>i=i+1 ;</p> <p>}</p> |
| <ul style="list-style-type: none"> • <u>C</u>as (choix) <u>d</u>ans <p>v1 : <u>d</u>ébut ;</p> <p>instruction 1 ;</p> <p>instruction n ;</p> <p><u>f</u>in ;</p> <p>v2 : <u>d</u>ébut ;</p> <p>instruction 1 ;</p> <p>instruction n ;</p> <p><u>f</u>in ;</p> <p><u>a</u>utres : instruction 1 ;</p> <p>instruction n ;</p> <p><u>f</u>in ;</p> <p><u>f</u>cas ;</p> | <ul style="list-style-type: none"> • switch (choix) { <p>case 1 : { instruction 1 ;</p> <p>instruction n ;</p> <p>break ;</p> <p>}</p> <p>case 2 : { instruction 1 ;</p> <p>instruction n ;</p> <p>break;</p> <p>}</p> <p>default : { instruction 1 ;</p> <p>instruction n ;</p> <p>break ;</p> <p>}</p> <p>}</p> |
| <ul style="list-style-type: none"> • <u>E</u>nt T[10] ; (tableau à une dimension) • <u>E</u>nt T[10][15] ; (tableau à deux dimensions) | <ul style="list-style-type: none"> • int T [10] ; • int T[10][15] ; |
| <ul style="list-style-type: none"> • <u>e</u>nt T[7]={1,2,5,9,4,2,4} ; | <ul style="list-style-type: none"> • int T[7]={1,2,5,9,4,2,4}; |

Programmation structurée

| | |
|--|---|
| <p>valeur n) : le type de la valeur à retourner (ex :ent) <u>début</u></p> <p style="padding-left: 40px;">instruction 1 ;</p> <p style="padding-left: 40px;">instruction n ;</p> <p style="padding-left: 40px;">retourne (y) ;</p> <p><u>fin</u> ;</p> | <pre>{ instruction 1 ; instruction n ; return (y) ; }</pre> |
| <ul style="list-style-type: none"> • FICHER * vehicule ; (declaration du fichier vehicule) | <ul style="list-style-type: none"> • FILE * vehicule ; |
| <ul style="list-style-type: none"> • OUVERTURE d'un FICHER et sa FERMETURE <u>Vehicule</u> ← ouvrir ("vehicule.dat", écriture) ; <p style="padding-left: 40px;">Instructions 1 ;</p> <p style="padding-left: 40px;">Instructions n ;</p> <p><u>Fermer</u> (vehicule);</p> | <ul style="list-style-type: none"> • Vehicule= fopen ("vehicule.dat", "w") ; <p style="padding-left: 40px;">Instruction 1 ;</p> <p style="padding-left: 40px;">Instructions n ;</p> <p style="padding-left: 40px;">fclose (vehicule) ;</p> |
| <ul style="list-style-type: none"> • ECRITURE <u>Ecrire</u> (&nom logique, veh) ; | <ul style="list-style-type: none"> • fwrite(&veh, sizeof(veh), 1, vehicule) ; |
| <ul style="list-style-type: none"> • LECTURE simple <u>Lire</u> (nom logique, veh) ; | <ul style="list-style-type: none"> • fread(&veh, sizeof(veh), 1, vehicule) ; |
| <ul style="list-style-type: none"> • LECTURE multiple Lire (nom logique, veh) ; <p><u>Tant que</u> nom fin de fichier(nom logique) <u>faire</u></p> <p style="padding-left: 40px;">Instruction 1 ;</p> <p style="padding-left: 40px;">Instruction n ;</p> <p style="padding-left: 40px;">Lire (nom logique, veh);</p> <p><u>Ftant que</u> ;</p> | <ul style="list-style-type: none"> • fread(&veh, sizeof(veh), 1, vehicule) ; <p style="padding-left: 40px;">while (!feof(vehicule)) ;</p> <pre>{ instruction 1 ; instruction n ; fread(&veh, sizeof(veh), 1, vehicule) ; }</pre> |
| <ul style="list-style-type: none"> • POSITIONNEMENT Positionner (nom logique, numéro enregistrement) ; <p>Instructions ;</p> | <ul style="list-style-type: none"> • fseek(vehicule, (long)(num enr – 1) * sizeof(veh) SEEK-SET) ; <p>Instructions ;</p> |

Exercice 1 :

```
/*
Ce programme permet de :
1) saisir deux entiers
2) calculer la somme, la différence, la multiplication, et la division des deux entiers
3) afficher les résultats des quatre opérations
*/
#include <cstdlib> /* On inclure la bibliothèque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliothèque des entrées/ sorties
*****/
using namespace std; // espace de nom
int main(int argc, char *argv[]) // le main
/* la fonction main avec ce prototype :
//////// int main(int argc, char **argv) //
permet de récupérer les éléments en ligne de comment.
argc : nombre de paramètres
argv : tableau contenant les paramètres
*/
{
    int var1, var2; // déclaration des variables de type entier
    cout<<"Veuillez saisir le 1er entier SVP"<<endl; /* affichage pour que l'utilisateur
saisisse les données*/
    cin>>var1; // la lecture du variable
    cout<<"Veuillez saisir le 2ème entier SVP"<<endl; /* affichage pour que l'utilisateur
saisisse les données*/
    cin>>var2; // la lecture du variable
    cout<<"La somme des entiers est : "<<var1+var2<<endl; /* faire l'addition des
variables et afficher la somme */
    cout<<"La différence entre les entiers est : "<<var1-var2<<endl; /* faire la différence
entre les deux variables et afficher la soustraction*/
    cout<<"La multiplication des entiers est : "<<var1*var2<<endl; /* faire la
multiplication des deux variables et afficher le produit */
    cout<<"La division des entiers est : "<<var1/var2<<endl; /* faire la division des deux
variables et afficher la division */
    system("pause"); // ce syntax permet de bloquer l'écran
}
}
```

Exercice 2 :

```
/*
Ce programme permet de faire la permutation cyclique dans le sens d'une aiguille de montre
Soit A,B,C,D 4 variables réelles
il faut donc remplir les 4 variables, on retiens la valeur de la 2ème variable dans une autre
nouvelle variable, ensuite, on retiens faire
la 1ère valeur dans la 2ème variable, la 2ème dans la 3ème, la 3ème dans la 4ème, la valeur
du nouvelle variable dans la 1ère - C'est ce
que l'on appelle la permutation - , en fin, on affiche le résultat
*/
#include <cstdlib> /* On inclure la bibliothèque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /* On inclure la bibliothèque des entrées/ sorties */
using namespace std; // espace de nom
```

Programmation structurée

```
int main(int argc, char *argv[]) //le MAINE
{
    float A,B,C,D,var1,var2; //déclaration des variables de type entier

    cout<<"Veuillez saisir la valeur de A SVP"<<endl; //affichage qui permet de saisir la
variable
    cin>>A; //la lecture
    cout<<"Veuillez saisir la valeur de B SVP"<<endl;
    cin>>B;
    cout<<"Veuillez saisir la valeur de C SVP"<<endl; //affichage qui permet de saisir la
variable
    cin>>C;
    cout<<"Veuillez saisir la valeur de D SVP"<<endl;
    cin>>D;
    /*******//
    E=A; //Pour conserver la valeur de A on affect sa valeur à la variable E
    A=D; //on affect la valeur D à la variable A
    D=C; //on affect la valeur C à la variable D
    C=B; //on affect sa valeur B à la variable C
    B=E; //on affect sa valeur E à la variable B
    cout<<"la nouvelle valeur de A apres purmutation A="<<A<<endl;
    cout<<"la nouvelle valeur de B apres purmutation B="<<B<<endl;
    cout<<"la nouvelle valeur de C apres purmutation C="<<C<<endl;
    cout<<"la nouvelle valeur de D apres purmutation D="<<D<<endl;
    /*******/
    system("pause"); /*bloquer l'écran & afficher un message fixe : " appuyer sur une touche
pour continuer ... "*/
}
```

Exercice 3 :

```
/*
Ce programme permet de :
- saisir 2 entiers
- comparer les deux valeurs des variables
- afficher le plus gran d'entre eux
+ si les valeurs des variables son egaux , on affiche un message de l'égalité
*/
#include <cstdlib> /*On inclure la bibliothèque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream
using namespace stdint main(int argc, char *argv[])
{
    int var1,var2; //Déclaration de deux variables
    cout<<"Veuillez saisir le 1ere entier SVP"<<endl;
    cin>>var1;
    cout<<"Veuillez saisir la 2eme entier SVP"<<endl;
    cin>>var2;
    if(var1<var2) //comparaison entre les 2 variables
    {
        cout<<"le plus grand entre les deux entier est : "<<var2<<endl; /*afficher le
plus grand des entiers*/
    }
    Else /*si la valeur du 1er variable n'est pas inferieur de l'autre on comparr les dans
les autres cas possible*/
    {
        if(var1>var2)
        {
```

```
        cout<<"le plus grand entre les deux entier est : "<<var1<<endl;
    /*afficher le plus grand des entiers*/
    }
    Else /*si la valeur de l'un est ni grand ni petite que l autre : alors les valeurs
son egaux*/
    {
        cout<<"les deux entiers sont egaux"<<endl; /*afficher que les entiers
saisie est egaux*/
    }
    }
    system("pause"); //pour bloquer l'écran
}
```

Exercice 4 :

/* Le programme qui permet de :

- saisir 3 entiers
 - comparer les trois valeurs des variables
 - afficher le plus grand d'entre eux
 - + si les valeurs des variables son égaux , on affiche un message de l'égalité
- */

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    int var1,var2,var3,max; //déclaration des variable de type entier
    bool egaux; //déclaration d un variable de type booléen
    cout<<"Veuillez saisir un entiers SVP"<<endl;
    cin>>var1;
    cout<<"Veuillez saisir un entiers SVP"<<endl;
    cin>>var2;
    cout<<"Veuillez saisir un entiers SVP"<<endl;
    cin>>var3;
    system("cls");
    cout<<endl<<endl;
    if(var1>var2) //tester si var1 est inferieur a var2
    {
        max=var1; //affectation du var1 au variable max
    }
    else
    {
        if(var2>var1) //tester si var2 est inferieur a var1
        {
            max=var2; //affectation du var1 au variable max
        }
        else
        {
            max=var1; //affectation du var2 au variable max
            egaux=true; //on retien la valeur vrai pour la variable égaux
        }
    }
    if(max>var3) //tester si max est inferieur a var3
    {
        cout<<"le plus grand est : "<<max<<endl<<endl<<endl; /*1 affichage du plus grand
des entiers */
    }
    else
    {
        if(max>var3) //tester si max est supérieur a var3
```

```

        {
            cout<<"la valeur la plus grande est : "<<max<<endl <<endl; /*/l
affichage du plus grand des entiers*/
        }
        {
            if(max<var3) //tester si la valeur de la variable égaux est vrai
            {
                cout<<"la valeur la plus grande est "<<var3; /*afficher que la
valeur maximale est var3*/
            }
            Else
            {
                if(egaux==true){
                    cout<<"les 3 valeur sont égaux" ; /*affichage que les 3
valeurs sont égaux */
                }
            }
        }
    }
    system("PAUSE");
}

```

Exercice 5 :

```

/*
le résoudre de l'équation  $ax+b=0$  ,a besoin d'un programme qui permet de :
+ Saisir deux valeurs ( le 1er est le coefficient du X ; et la 2eme c'est le terme
indépendante)
+ tester le 1er valeur :
- si il est différente a 0 ( on fait l opération Appropriées a ce condition)
- sinon on faire un autre teste pour le terme indépendante ,si est égale a 0 ( on fait
l opération appropriées a ce condition)
-- sinon on affiche juste un message
+ a chaque fois on affiche un message ça depend au saisi de l'utilisateur
*/
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    float a,b,c; //déclaration des variables de type réel
    cout<<"\t ax+b=0"<<endl<<endl;
    cout<<"Veuillez SVP saisir la valeur de a : "; /*affichage qui demande a l'utilisateur
de saisir la valeur de la coefficient de X*/
    cin>>a;
    cout<<"Veuillez SVP saisir la valeur de b : "; /*affichage qui demande a l'utilisateur
de saisir la valeur du terme indépendante*/
    cin>>b;
    if(a!=0 & b!=0) /*on teste si la valeur de a est différent a 0 ( != c'est le signe différent
en c++*/
    {
        c=-b/a; // faite la calcule
        cout<<"la solution est : "<<c<<endl;
    }
    else
    {
        if(b= =0) // on teste si la valeur de b est égale a 0
        {

```

```
        cout<<"Toute possibilite"<<endl;
    }
    else
    {
        cout<<"Pas de solution"<<endl;
    }
}
system("PAUSE");
}
```

Exercice 6 :

/*
le résoudre de l'équation $ax^2+bx+c=0$,a besoin d'un programme qui permet de :
+ Saisir trois valeurs (le 1er est le coefficient du X^2 , le 2eme est le coefficient du X ;
et la 3eme est le terme indépendante)
+ tester le 1er valeur
- si il est différente a 0 (on fait l opération Appropriées a l'equation du 1ere degre)
- sinon on faire la calcule de delta
-- si delta est supérieur a 0, donc on calcule le $X1$ et le $X2$ (le même méthode que
le math $(-b-\sqrt{\text{delta}})/2*a$).
-- si delta est inférieur a 0, on affiche un message
-- si delta st egal a 0 , on calcule juste $X1$
+ a chaque fois on affiche un message ça dépend au saisi de l'utilisateur
*/

```
#include <cstdlib>
#include <iostream>
#include <math.h> /*on inclure " math.h " pour travailler avec des opération mathématique
comme le racine carrée*/
using namespace std ;
int main(int argc, char *argv[])
{
    int a,b,c; //déclaration des variables de type entier
    float X1,X2,delta; //déclaration des variables de type réel
    cout<<"Veuillez saisir le coeficient du X2 : ";
    cin>>a;
    cout<<"Veuillez saisir le coeficient du X : ";
    cin>>b;
    cout<<"Veuillez saisir le term independant : ";
    cin>>c;
    if(a==0) //on teste si la valeur du a est égale a 0
    {
        X1=-c/b; //on fait la calcule
        cout<<"Equation du 1ere degre."<<endl;
        cout<<"la valeur de x: "<<X1<<endl;
    }
    else
    {
        cout<<"delta =(b*b)-4*c*a"<<endl;
        delta=(b*b)-(4*c*a); //on fait la calcule du delta
        if(delta>0) //on teste si la valeur de delta est inferieur a 0
        {
            cout<<"la valeur de delta >0"<<endl; //afficher la valeur du delta
            X1=(-b-sqrt(delta))/(2*a); //calculer X1
            cout<<" X1 = "<<X1<<endl; //afficher la valeur du X1
            X2=(-b+sqrt(delta))/(2*a); //calculer X2
            cout<<" X2 = "<<X2<<endl; //afficher la valeur du X2
        }
    }
    else
}
```

```
        { if(delta<0) //on teste si la valeur de delta est superieur a 0
          {
            cout<<"l'ensemble des solutions vide"<<endl; //affichage du message
          }
          else
          { X1=-b/(2*a); //claculer X1
            cout<<"La valeur de X1 : "<<X1<<endl; //afficher le X1
          }
        }
    }
}
system("pause");
}
```

Exercice 7 :

/*

le programme suivante faire le calcul de la participation au pris du repas de ses employés de la façon suivanate:

- +si l'employer est celibataire la participation est de 20%
- +si l'employer est marier la participation est de 25%
- si il a des enfants la participation est de 10%
- + la participation est plafonné a 50%
- si le salaire mensuelle est inferieur a 6000 Dhs la participation est majoré de 10%
- +a chaque fois quand a le résultat de la participation on affiche le

*/

```
#include <cstdlib>
#include <iostream>
#include <string> //on inclure la classe string pour travaillée avec les chaine de caractère
using namespace std;
int main(int argc, char *argv[])
{
    int salaire,X; //declaration des variables de type entier
    float participation; //declaration des variables de type reel
    string nom,situation; //declaration des variables de type chaine de caractaire

    cout<<"Veuillez SVP saisir votre nom"<<endl; //affichage du saisie
    cout<<"Mon nom est: ";
    cin>>nom; //la lecture de saisie
    cout<<"Veuillez SVP saisir votre situation"<<endl; //affichage du saisie
    cout<<"Ma situation est: ";
    cin>>situation; //la lecture de saisie
    cout<<"Veuillez SVP saisir votre salaire en DH"<<endl; //affichage du saisie
    cout<<"Mon salaire est: ";
    cin>>situation; //la lecture de saisie
    if(situation=="celibataire") //tester si la situation du employier est clibataire
    {
        participation=20;
    }
    else{
    cout<<"Veuillez SVP saisir le nombre de vos enfants"<<endl; //affichage du saisie
    cout<<"le nombre de mes enfants est =";
    cin>>X; //la lecture de saisie
    participation=(25+(X*10)); //calculer la participation
    if(participation>50){ //Un test sur la valeur de la participation
        participation=50;
    }
    }
    if(salaire<6000){ //tester si le salaire est inferieur à 6000 dhs
```

Programmation structurée

```
        participation=participation+(participation*10)/100;
    }
    cout<<"votre participation Mr/Mm"<<nom<<"est de "<<participation<<"%"<<endl;
//affichage de valeur final de la participation
    system("pause");
}
```

Exercice 8 :

```
/*
ce programme faite pour que soit un étudiant ayont cinq notes dans l'examen.
donc on faite la calcul de la moyenne de cette etudiant sachant que toutes les notes on le
meme coefficients
alors on demande a l'utilisateur d'entrer les notes pour que calculer la moyenne general et
on affichent le resultat
*/
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/
sorties *****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le main
{
    int i; //declaration d un entier
    float note , somme; //declaration des variables de types reel
    somme =0; //inistatisation par 0
    for(i=0; i<5; i++) //la boucle pour
    {
        cout<<"Veillez saisir une note SVP : "; //affichage pour faire la saisi des notes
        cin>>note; //la lecture des notes
        somme = note+somme; //calculer la somme des notes
    }
    note = somme/5; //calculer la moyenne des notes
    cout<<"la moyene est : "<<note<<endl; //afficher la moyenne general
    system("PAUSE"); //pour bloquer l ecran
}
```

Exercice 9 :

```
/* Ce programme calcul et affiche la table de multiplication d'un entier saisi par l'utilisateur
*/
#include <cstdlib> /* On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/
sorties *****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le main
{
    int n,i,produit; //declaration d un entier
    cout<<"veuiller SVP saisir la valeur de votre entier n= "; //affichage du saisi
    cin>>n; //la lecture du saisi
    for(i=0;i<=10;i++){
        produit=n*i; //calculer du produit de n du 0 jusqu'à 10
        cout<<n<<"*"<<i<<"="<<produit<<endl; //affichage du resultat
    }
    system("pause");//pour bloquer l ecran
    return 0;
}
```

Exercice 10 :

```
/* le programme suivante permet de :
    + calculer la somme de la suite S=1+2+3+.....+n
    - on calcule la somme des chiffres jusqu'a l'element saisite par l'utiliisateure
    + afficher le resultat de la somme de la suite
*/
#include <cstdlib> /* On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /***** On inclure la bibliotheque des entés/
sorties *****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le main
{
    int somme, suite , i; //declaration des variables de type entier
    somme =0; //initialisation de la variable somme par 0
    cout<<"veuillez saisir le nombre de la suite : "; //affichage de saisie
    cin>>suite; //la lectuer de saisie
    for(i=0 ;i<=suite;i++) //boucle pour
    {
        somme=somme+i; // calculer la somme de la suite
    }
    cout<<"La somme de la suite "<<suite<<" est : "<<somme<<endl; //afficher le resultat de
la suite
}
```

Exercice 11 :

```
/*le programme suivante permet de :
    + calculer la somme de la suite S=1-2+3-4.....(+)-n
    - donc la suite contienne des chiffres paire et autre impaire, c'est pour cela on fait le
calcul de la somme
    des chiffres paire et aussi la somme des chiffres impaire jusqu'a l'element saisite
par l'utiliisateure
    - après on fait la difference entre la somme des chiffres paire et impaire
    + afficher le resultat de la suite */
```

Version1 :

```
#include <cstdlib> /* On inclure la bibliotheque qui sert principalement à déclarer la
fonction C system()*/
#include <iostream> /***** On inclure la bibliotheque des entés/
sorties *****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le main
{
    int n, x,i , somme; //declaration des variable de type entier
    somme=0; //inisialization du somme par 0
    cout<<"veuillez saisir un entier SVP : "; //affichage de saisie
    cin>>n; //la lecture de saisie
    for(i=1;i<=n;i++) //boucle pour
    {
        x=i%2;
        if(x==0){
            somme = somme-i; //calculer la somme des entier paire
        }
        else{
            somme = somme+i; //calculer la somme des entier impaire
        }
    }
    cout<<" la somme est : "<<somme<<endl; // Affichage du resultat de la somme
}
```

Version 2:

```
#include <cstdlib> /* On inclure la bibliotheque qui sert principalement à déclarer la
fonction C system()*/
#include <iostream> /***** On inclure la bibliotheque des entés/
sorties *****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le main
{
    int n,i , somme1,somme2; //declaration des variable de type entier
    somme1 =0; //inisialization du somme 1 par 0
    somme2 =0; //inisialization du saomme 2 par 0
    cout<<"veuillez saisir un entier SVP : "; //affichage de saisie
    cin>>n; //la lecture de saisie
    for(i=1;i<=n;i+=2) //boucle pour calculer la somme des entier paire
    {
        somme1 = somme1+i;
    }
    for(i=0;i<=n;i+=2){ // Une boucle pour calculer la somme des entier impaire
        somme2=somme2+i;
    }
    somme=somme1-somme2; //calculer la difference entre la somme des paire et la
somme des impaire
    cout<<" la somme est : "<<summe1-somme2<<endl ; //On affiche le resultat
}
}
```

Exercice 12 :

```
/* le factoriel d'un entier  $N! = N*(N-1)*(N-2)*\dots*1$ 
le programme ont pour fonction picipale de faire le calcule de cette factoriel N !
donc on demande a l'utilisateur de saisir un entier pour faire le calcule ;
on travail sur ce entier ( :la multiplication des nombres a partir de '1' jusqu'a l'element
saisie par l'utilisateur);
et enfin on affiche le resultat */
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /***** On inclure la bibliotheque des entés/
sorties *****/

using namespace std;
int main(int argc, char *argv[])
{
    int N,Var1,i; //declaration des variables de type entier
    Var1=1; //inisialisation du variable Var1 par 1
    cout<<"veuillez saisir un entier SVP "; //affichage de saisie
    cin>>N; //lecture de saisie
    for(i=1; i<=N ; i++) //boucle pour
    {
        Var1=i*Var1 ; //calculer la factoriel du N qui est saisite par l utilisateur
    }
    cout<<"le factoriel "<<N<<"! est: "<<Var1<<endl; //afficher le factoriel du N
}
}
```

Exercice 13 :

```
/* ce programme faite pour que soit un étudiant ayont X notes dans l'examen.(nombre de
notes de l'etudiant n'est pas connu)
donc on faite la calcule de la moyenne de cette etudiant,
sachant que toutes les notes ont des coeficients differents ;
alors on demanade a l utilisateur d'entrer les notes pour que
calculer la moyenne general ( pour ça on est besoin de calculer
la somme des coeficients a part et ainsi la somme des notes multiplier par leurs coeficients )
```

```
a la fin , on affichent le resultat de la moyenne */
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/
sorties *****/
using namespace std;
int main(int argc, char *argv[])
{
    int coef, somme1,i; //declaration des variables de type entier
    float Note , N , somme2,res ; //declaration des variables de type reel
    cout<<"veuillez saisir le nombre des notes SVP : "; /*affichage de saisie du nombre
des notes pour gere l arret de les boucles suivantes*/
    cin>>N; // lecture de N
    somme1=0; //inialization de la variables somme1 par 0
    somme2=0; //inialization de la variables somme2 par 0
    i=1; //inialization de la variables i par 1
    while(i<=N) //la boucle Tant que
    {    cout<<"Veuillez saisir une note : "; //affichage de saisie d une note
        cin>>Note; // la lecture du note
        cout<<"Veuillez saisir le coeficient de cette note : "; /*affichage de saisie d un
coeficient de cette note */
        cin>>coef; //la lecture du coeficient
        somme1 = somme1+coef; //calculer la somme des coeficients
        somme2=somme2+(coef *Note); /*calculer la somme des coeficients
multiplier par la note */
        i++; //gere le compteur
    }
    res = somme2/somme1 ; //calculer la moyenne des notes
    cout<<"La moyenne est : "<<res<<endl; // afficher le resultat ( la moyenne des notes)
    system("PAUSE");
}
```

Exercice 14 :

```
/* Ce programme affiche le maximum , le minimum , la somme , et la moyenne des entiers
saisite par l utilisateur
Au debut , on demande a l'utilisateur de saisir un entier pour que on debute la comparaison
avec les entiers suivantes, ensuite on teste si la valeur suivante est plus grand que le
precedente , on prend le dans un variable qui correspondante ;sinon on teste si il est plus petit,
et aussi en le prend dans le variable qui correspondante
A chaque fois ,on faite la somme des entiers , enfin on calcule la moyenne de ces entiers et
on l'affiche le resultat */
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/
sorties *****/
#include <string> // on inclure la classe string pour travailler sur chaine de caractaire
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le main
{
    int var1,max, min,j, i,somme; //Déclaration des variables de type entier
    float moy; //declaration d un variable de type reel
    string rep; //declarationn d'un variable de type chaine de caractere
    i=1; //initialisation du compteur par un 1
    cout<<"element "<<i<<" : "; //affichage de saisir
    cin>>var1; //lecture de l'écriture et placé dans la variable var1
    max = var1; //*****//
    min = var1; // initialisation du (min , max, et somme par var1) //
    somme = var1; //*****//
    do
```

```

{ cout<<"element "<<i+1<<" : "; //affichage de saisir
  cin>>var1; //lecture du contenu de la variable var1
  if(max < var1) //comparéson entre les 2 varibales
  { max = var1; //affectation du contenu du variable var1 au variable max
  }
  Else /* si la valeur du 1er variable n'est pas inferieur de l'autre on compar les
dans les autres cas possible*/
  {      if(min > var1) //comparéson entre les 2 varibales
        {
          min = var1 ; //affectation du contenu du variable var1 au variable min
        }
      }
  }
  somme = somme+var1; //calculer la somme des entiers saisie
  i++; //Incrémenter le compteur
  cout<<"Voulez-vous continuez ?(oui/non)  "; /*demander a l'utilisateur de saisir
"oui" si il veut resaisir un autre entier */
  cin>>rep; //la lecture de la reponse
  }while(rep == "oui");
  moy =somme/i-1; //calculer la moyenne
  cout<<"le maximum des entiers est : "<<max<<endl;
  cout<<"le minimum des entiers est : "<<min<<endl;      //    A f f i c h a g e      //
  cout<<"la somme des entiers est : "<<somme<<endl;      //    d u   R e s u l t a t      //
  cout<<"la moyenne est : "<<moy ;
}

```

Exercice 15 :

/*le programme permet de :

- + demander a l'utilisateur d'entrer le code
- + demander a l'utilisateur d'entrer le nom
- + demander a l'utilisateur d'entrer le total de la consommation d'électricité par mois
- + Calculer le montant Hortax par tranche
- + Calculer le montant TTC par tranche
- + Calculer le montant Totale de la facture plus les frais fixe d'abonnement : 70 Dh

(si la consommation est inferieur de 110 , on fait la calcule du montant hortaxte, , TTC et totale pour just une seul tranche , si la consommation est superieur a 110 et inferieur a 220 ;on fait la calcule du montant hors taxe , TTC et totale pour le prmeier et la deuxieme tranche , mais quand la consommation est superieur de 220 , on fait la calcule du montant hortaxte, TTC et totale pour le premier , la deuxieme et la troisieme tranche)

- + A chaque fois ,on fait l'affichage du resultat avec le nom ,le code et toutes les information */

```

#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sorties *****/
#include <string> // on inclure la bibliotheque du chaine de caractere
using namespace std; //espace de nom
int main()
{
  float conso , hort1, hort2, hort3, ttc1, ttc2, ttc3, tot; //declaration des variables de types réel
  int code; //declaration d'une variable "Code" de type entier
  string nom,rep; //declaration des variables de type chaine de caractère
  do //faire
  {
    system("cls"); //pour vider l'écran
    cout<<"veuillez saisir votre Nom : "; //affichage pour saisir le nom
    cin>>nom; //la lecture de nom
    cout<<"veuillez saisir votre Code : "; //affichage pour saisir le code
    cin>>code; //la lecture de code

```

Programmation structurée

```
cout<<"veuillez saisir votre consommation : "; //affichage pour saisir la consommation
cin>>conso; //la lecture de consommation
if(conso <=110) //tester si la consommation est inferieur de 110
{
    system("cls"); //pour vider l'ecran
    cout<<"nom : "<<nom<<"\t Code : "<<code<<endl<<endl<<"\t\t la consommation
:"<<conso<<" Dh"<<endl ;
    hort1= conso*0.9; //calculer le hors taxe
    cout<<"le mantont hortaxt est : "<<hort1<<" Dh"<<endl; /*afficher le mantont du hors
taxe */
    ttc1 = hort1+(hort1*0.2); //calculer le TTC du 1er Tranche
    cout<<"le montant TTC est : "<<ttc1<<" Dh"<<endl; //afficher le montant du TTC
    tot= ttc1+70; //calculer le totale de la facture
    cout<<"le totale de la facture est : "<<tot<<" Dh"<<endl; //afficher le totale de la facture
}
Else //si la consommation est supperieur de 110
{
    if(conso >110 && conso<220) /*tester si la consommation est supperieur a 110 et
inferieur a 220*/
    {
        system("cls"); //pour vider l'ecran
        cout<<"nom : "<<nom<<"\t Code : "<<code<<endl<<endl<<"\t\t la
consommation : "<<conso<<" Dh"<<endl<<endl;
        hort1= 110*0.9; //calculer le hortaxt du 1er Tranche
        cout<<"le 1er mantont hortaxt est : "<<hort1<<" Dh"<<endl; /*afficher le
mantont du 1er hors taxe */
        hort2= (conso-110)*0.98; //calculer le hortaxt du 2eme Tranche
        cout<<"le 2eme mantont hortaxt est : "<<hort2<<" Dh"<<endl; /*afficher
le mantont du 2eme hors taxe */
        ttc1 = hort1+(hort1*0.2); //calculer le TTC du 1er Tranche
        cout<<"le 1er montant TTC est : "<<ttc1<<" Dh"<<endl; /*afficher le 1er
montant du TTC*/
        ttc2 = hort2+(hort2*0.2); //calculer le TTC du 2eme Tranche
        cout<<"le 2eme montant TTC est : "<<ttc2<<" Dh"<<endl; /*afficher le
2eme montant du TTC */
        tot= ttc1+ttc2+70; //calculer le totale de la facture
        cout<<"le totale de la facture est : "<<tot<<" Dh"<<endl; /*afficher le totale
de la facture */
    }
    Else //si la consommation est supperieur de 220
    {
        system("cls"); //pour vider l'ecran
        cout<<"nom : "<<nom<<"\t Code : "<<code<<endl<<endl<<"\t\t la
consommation : "<<conso<<" Dh"<<endl<<endl;
        hort1= 110*0.9; //calculer le hortaxt du 1er Tranche
        cout<<"le 1er mantont hortaxt est : "<<hort1<<" Dh"<<endl; /*afficher le
montant du 1er hors taxe */
        hort2= 110*0.98; //calculer le hortaxt du 2eme Tranche
        cout<<"le 2eme mantont hortaxt est : "<<hort2<<" Dh"<<endl; /*afficher
le montant du 2eme hors taxe */
        hort3= (conso-220)*1.20; //calculer le hortaxt du 3eme Tranche
        cout<<"le 2eme mantont hortaxt est : "<<hort3<<" Dh"<<endl; /*afficher
le montant du 3eme hors taxe */
        ttc1 = hort1+(hort1*0.2); //calculer le montant du 1er TTC
        cout<<"le 1er montant TTC est : "<<ttc1<<" Dh"<<endl; /*calculer le TTC
du 1er Tranche*/
        ttc2 = hort2+(hort2*0.2); //calculer le montant du 2eme TTC
```

```

        cout<<"le 2eme montant TTC est : "<<ttc2<<" Dh"<<endl; /*calculer le
TTC du 2ème Tranche*/
        ttc3 = hort3+(hort3*0.2); //calculer le montant du 3ème TTC
        cout<<"le 2eme montant TTC est : "<<ttc3<<" Dh"<<endl; /*calculer le
TTC du 3ème Tranche*/
        tot= ttc1+ttc2+ttc3+70; //calculer le mantant totale de la facture
        cout<<"le totale de la facture est : "<<tot<<" Dh"<<endl; /*afficher le
totalte de la facture */
    }
}
    cout<<"voulez vous continuez (oui/non)?"<<endl; /*affichage pour saisir si il veux
continuer taper oui sinon taper non*/
    cin>>rep; //lire la reponse de saisie
}while(rep=="oui"); //tester si la reponse est existe
}

```

Exercice 16 :

```

#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/
sorties *****/
#include <string> //on inclure la bibliotheque du chaine de caractaire
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le main
{
    bool sortire ; //declaration d une variable de type booleen
    int jourN,jourA,moisA,moisN,AnneN,anneeA,age; //declaration des variables de type entier
    string rep,nom,prenom; //declaration des variables de type chaine de caractaire
    rep="non"; //inialisisation du variable "rep" par la valeur non
    sortire = false ; //intialisation du variable "sortire " par la valeur faux
do
{
    cout<<"Nom : "; //affichage pour saisie du nom
    cin>>nom; //lecture du nom
    cout<<"Prenom : "; //affichage pour saisie du prenom
    cin>>prenom; //lecture du prenom
    do
    {
        cout<<"Jour de naissance : "; //affichage pour saisir le jour de naissance
        cin>>jourN; //lire le jour de naissance
        if(jourN<1 || jourN>31) /*Si N° 1 *** tester si le jour de naissance est superieur a 1 et
inferieur a 31*/
        {
            cout<<"Erreur (La date du jour est invalide) . Voulez-vous recomencez (oui/non)?"
/*afficher un erreur si la date qui l'utilisateur a saisie est invalide*/
            cin>>rep; //lire la reponse de resaisire
        }
        Else
        {
            Do
            {
                cout<<"Mois de naissance : "; //affichage pour saisir le mois de naissance
                cin>>moisN; //lire le mois de naissance
                if(moisN<1 || moisN>12)
                {
                    cout<<"Erreur (La date du mois est invalide).Voulez-vous recomencez (oui/non)?"
//afficher un erreur si la date qui l'utilisateur a saisie est invalide
                    cin>>rep; //lire la reponse de resaisire
                }
            }
        }
    }
}

```

```

Else /      {
  Do
  {
    cout<<"Annee de naissance : "; //affichage pour saisir l annee de naissance
    cin>>AnneN; //lire l annee de naissance
    if(AnneN>2010)
    {
      cout<<"Erreur (La date d'annee est invalide) . Voulez-vous recommencez
(oui/non)?"; //afficher un erreur si la date qui l'utilisateur a saisie est invalide
      cin>>rep; //lire la reponse de resaisire
    }
    Else
    {
      Do
      {
        cout<<"Jour Actuelle : "; //affichage pour saisir le jour actuelle
        cin>>jourA; //lire le jour actuelle
        if(jourA<1 || jourA>31)
        {
          cout<<"Erreur (La date du jour est invalide) . Voulez-vous recommencez
(oui/non)?"<<endl; //afficher un erreur si la date qui l'utilisateur a saisie est invalide
          cin>>rep; //lire la reponse de resaisire
        }
        else
        {
          Do
          {
            cout<<"Mois actuelle : "; //affichage pour saisir le mois  actuelle
            cin>>moisA; //lire le mois actulle
            if(moisA<1 || moisA>12) //SI N°6
            {
              cout<<"Erreur (La date du jour est invalide) . Voulez-vous recommencez
(oui/non)?"<<endl; //afficher un erreur si la date qui l'utilisateur a saisie est invalide
              cin>>rep; //lire la reponse de resaisire
            }
            else
            {
              Do
              {
                cout<<"Annee actuelle : "; //affichage pour saisir l annee  actuelle
                cin>>anneeA; //lire l annee actuelle
                if(anneeA <=2009)
                {
                  cout<<"Erreur (La date du jour est invalide) . Voulez-vous recommencez
(oui/non)?"<<endl; //afficher un erreur si la date qui l'utilisateur a saisie est invalide
                  cin>>rep; //lire la reponse de resaisire
                }
                else
                {
                  age = anneeA-AnneN; /*calculer la difference entre l'annee actuelle et
l'annee de naissance */
                  cout<<endl<<"Bonjour " <<nom<<" " <<prenom<<" vous avez
" <<age<<" ans." <<endl; //afficher le message avec un mon , prenom , et age de l utilisateur
                  if(jourA == jourN && moisA == moisN) //si n:8 *****
                  {

```

```
        cout<<endl<<endl<<"*****Joyeux
anniversaire*****" <<endl<<endl; /*afficher un message pour dire a l
utilisateur Joyeux anniversaire */
    }
    rep=="oui";
    sortire = true;
}
}while(rep=="oui" && sortire == false);
}
}while(rep=="oui" && sortire == false );
}
}while(rep=="oui" && sortire == false);
}
}while(rep=="oui" && sortire == false );
}
}while(rep=="oui" && sortire == false );
}
}while(rep=="oui" && sortire == false);
cout<<"voulez-vous continuez (oui/non)?"; /*affichage pour resaisire autres informations
avec une valeur : "oui" ou bien "non"*/
cin>>rep; //lire la reponse de saisie
}while(rep=="oui");
system("PAUSE");
}
```

Exercice 17 :

/*Ce programme permet de lire une suite N entiers dans un tableau
d'abord, on donne la possibiliter a l utilisateur de saisir des entiers pour remplir le tableau
dans ce tableau on doit lire un nombre entier comme donnée et chercher si se nombre existe
* si existe on affiche un message
* si n'existe pas on affiche un autre message */

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/
sorties *****/
using namespace std;
int main(int argc, char *argv[])
{
    int Tent[10]; /*declaration d un tableau qui est nomer par " Tent " et on donne la taille du
tableau : 10 case */
    int i, entier ; //declaration des variables de type entier
    bool trouve; //declarer une variable "trouve" de type booleen
    trouve = false; //inisialisation du variable "trouve" par faux
    for(i=0; i<=9;i++) //la boucle pour
    {
        cout<<"Veuillez saisir un entier SVP : "; //affichage de saisie pour remplir le tableau
        cin>>Tent[i]; //lire les variables et mettent chaque variable dans un case
    }
    cout<<"veuillez entrer un entier SVP: "; //afficher de saisie un entier
    cin>>entier; //lire
    i=0;
    do
    {
        if(Tent[i]== entier ) //tester si l'entier saisie est dans le tableau
        {
            cout<<"L'entier est existe : " <<endl; /*affichage pour dire que l'entier est
existe deja dansle tableau */
```

```

        trouve = true;
    }
    i++;
}while(i!=0 && trouve == false);
if(trouve == false)
{cout<<"l'element n'existe pas"<<endl;
}

```

Exercice 18 :

```

#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/
sorties *****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le main
{
    int i,j, min, max, moy, posmin, posmax; //declaration des variables de type entier
    int Tab[50]; // declaration d un tableau de type entier avec une taille de 10
    posmin = 0; // initialisation par 0
    posmax = 0; // initialisation par 0
    i=0; // initialisation par 0
    do
    {
        cout<<"Element"<<i+1<<" : "; //oooooooooooooooooooooooooooooooooooooooooooo//
        cin>>Tab[i]; // Remplire le tableau //
        i++; // //
    }while(Tab[i-1]!=0 && i!=50); // //
    if(i==50) // tester si la valeur de i est egale a 10
    {
        i++;
    }
    max=Tab[0]; ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    min= Tab[0]; // Initialisation du Min, Max et moy par le 1er //
    // valeur saisit //
    moy= Tab[0]; ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    for(j=1; j<=i-2; j++)
    {
        if( min>Tab[j]) //tester si la nouvelle valeur est plus petit que la precedante
        {
            min= Tab[j]; //affectation de la valeur de Tab[j] au variable min
            posmin = j; //affectation pour prendre la position du min
        }
        else
        {
            if(max < Tab[j]) //tester si la nouvelle valeur est plus grand que la precedante
            {
                max = Tab[j]; //affectation de la valeur de Tab[j] au variable max
                posmax =j; //affectation pour prendre la position du max
            }
        }
        moy=moy+Tab[j]; //la calcule de la moyenne des entiers
    }
    //***** Affichage du resultat obtenu *****
    cout<<"le maximum des entiers est : "<<max<<" sa position est :
"<<posmax+1<<endl;
    cout<<"le minimum des entiers est : "<<min<<" sa position est : "<<posmin+1<<endl;
    cout<<"la moyenne des entiers est : "<<moy/(i-2)<<endl;
}

```

Exercice 19 :

/* Le programme suivante permet de saisir une liste de classe comprenet le nom de chaque stagiere et sa moyenne generale, ainsi il permet d'afficher les noms et les moyennes de tous les stagieres qui on une moyenne superrieur a la moyenne de la classe , plus que ça le programme doit afficher le nom et la moyenne du premier et du dernier de la classe donc on remplit un tableau a partir du saisi de l'utilisateur ;
on faite la clacule de la moyenne general de toute les notes saisie et on affiche
on cherche les notes qui on plus grand et plus petit que la moyenne , c'est pour ca on tester si une note est superrieur ou inferieur a la moyenne on affiche avec le nom qui correspondant.*/

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sorties *****/
#include <string> //on inclure la classe string pour travailler sur les chaine de caractaire
using namespace std; //espace de nom
int main(int argc, char *argv[]) // le main
{
    string Tnom[5]; // declaration d un tableau de type chaine de caractere
    float Tmoy[5]; //declaration d'un tableau de type réel
    int i; //declaration d un variable de type entier
    float GrandM ,PetitM,S; //declaration des variables de type réel
    string diligent , paresseux; //declaration d un variable de type chaine de caractare
    S=0; // intialisation par 0
    for(i=0; i<=4; i++)
    {
        cout<<endl<<endl; //-----//
        cout<<"n:"<<i+1<<endl; // Affichage //
        cout<<"Nom :"; // de saisit //
        cin>>Tnom[i]; // pour remplire //
        cout<<"Sa moyenne: "; // le tableau //
        cin>>Tmoy[i]; //-----//
        S=S+Tmoy[i]; //calculer la somme des entiers
    }
    S= S/5; //calculer la moyenne
    system("cls"); // vider l'ecran
    cout<<"Qui on eu une note supperieur a la moyenne general : "<<endl;
    for(i=0; i<=4; i++)
    {
        if(S<Tmoy[i]) //tester si la note est superrieur a la moyenne de la classe
        {
            cout<<"\t "<<Tnom[i]<<" : "<<Tmoy[i]<<endl<<endl;
        }
    }
    GrandM=Tmoy[0]; //initialisation du variable par la 1er moyenne
    PetitM=Tmoy[0]; //initialisation du variable par la 1er moyenne
    diligent=Tnom[0]; //initialisation du variable par le nom la 1er moyenne
    paresseux =Tnom[0]; //initialisation du variable par le nom la 1er moyenne
    for(i=0; i<=5; i++)
    { // pour trouver le premier de la classe
        if(GrandM < Tmoy[i])
        {
            GrandM= Tmoy[i];
            diligent= Tnom[i];
        }
    }
    else
    { //pour trouver le dernier de la classe
```

```

    if(PetitM >Tmoy[i])
    {
        PetitM = Tmoy[i];
        paresseux = Tnom[i];
    }
}
}
/* ***** Affichage du resultat *****/
cout<<"\t *****"<<endl<<endl;
cout<<"# Le 1ere de la classe est :"<<diligent<<" ,sa moyenne est :
"<<GrandM<<endl<<endl;
cout<<"# Le dernier de la classe est :"<<paresseux<<" ,sa moyenne est :
"<<PetitM<<endl<<endl<<endl;
cout<<"\t *****"<<endl<<endl;
system("PAUSE");
}

```

Exercice 20 :

```

#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/
sorties *****/
#include <string> // on inclure la classe string pour travailler sur des chaine de caractere
using namespace std;//espace de nom
int main(int argc, char *argv[])
{
    int place,i,choix,choix1; //declaration des variables de type entier
    string rep,nom,siti; //declaration des variables de type chaine de caractere
    int Tnum[6]; //declaration d'un tableau de type entier avec une taille de 6
    string Tnom[6]; //declaration d'un tableau de type chaine de caractere avec une taille de 6
    choix = 0; //initialisation par 0
    choix1=0; //initialisation par 0
    i=0; //initialisation par 0
do
{
    cout<<"\t Menu du Wagon "<<endl;
    cout<<"Pour Reserver une place .Appuyer sur 1"<<endl; /*    L E          */
    cout<<"Pour Supprimer une place.Appuyer sur 2"<<endl; /*          M E N U      */
    cout<<"Pour Quitter .Appuyer sur 3"<<endl; /*          _____ */
    cin>>choix;
    switch(choix)
    { case 1:
        {
            cout<<"Voulez-vous entrer 'OUI' si le voyageure est fume "; /*demande a
l'utilisateur de saisir si le voyageur fume*/
            cin>>siti; // lire la reponse
            if(siti=="oui") //si la situation est oui
            {
                do
                {
                    if(i<3)
                    {
                        if(Tnom[i]==" " && Tnum[i]==0)
                        {
                            cout<<"Veuillez saisir un nom "; /*demande a l'utilisateur de saisir le nom du voyageur*/
                            cin>>Tnom[i]; //lire le nom
                            cout<<"Veuillez saisir le num de la place "; /*demande a l'utilisateur de saisir le
numero de la place */

```

```
        cin>>Tnum[i]; //lire le numero du place
    }
    else
    {
        cout<<"Veuillez saisir un nom "; //demande a l'utilisateur de saisir le nom
        cin>>Tnom[i]; //lire le nom
        cout<<"Veuillez saisir le num de la place "; /*demande a l'utilisateur de
saisir le numero */
        cin>>Tnum[i]; // lire le numero
    }
    }
    i++; // Incrementation du compteur
    cout<<"voulez vous faire une autre reservation \t"; /*demande a l'utilisateur
de saisir si il veut faire une autre reservation*/
    cin>>rep; //lire la reponse
    }while(i<3 && rep=="oui");
    if(i>=3)
    {
        cout<<" Y a aucune place vide"; // si il dépassé le numero de la place 3
    }
    }
    else
    {
        i=3;
        do
        {
            if(i>=3)
            {
                if(Tnom[i]==" " && Tnum[i]==0)
                {
                    cout<<"Veuillez saisir un nom "; /*demande a l'utilisateur de saisir le nom
du voyageur*/
                    cin>>Tnom[i]; //lire le nom
                    cout<<"Veuillez saisir le num de la place "; /*demande a l'utilisateur de
saisir le numero de la place */
                    cin>>Tnum[i]; //lire le numero du place
                }
            }
            else
            {
                cout<<"Veuillez saisir un nom "; /*demande a l'utilisateur de saisir le
nom du voyageur*/
                cin>>Tnom[i]; //lire le nom
                cout<<"Veuillez saisir le num de la place "; /*demande a l'utilisateur de
saisir le numero de la place */
                cin>>Tnum[i]; //lire le numero du place
            }
        }
        }
        i++;
        cout<<"voulez vous faire une autre reservation \t"; /*demande a l'utilisateur
de saisir si il veut faire une autre reservation*/
        cin>>rep; //lire la reponse
        }while(i<6 && rep=="oui");
        if(i>=6)
        {
            cout<<" Y a aucune place vide"; // si il dépassé le numero de la place 6
        }
    }
}
```

```
    }
    case 2:
    {
        if(choix == 2)
        {
            /*      M E N U          */
            /* de la suppression */
            cout<<"Voulez-vous supprimez a partire du nom.....(3)"<<endl;
            cout<<"Voulez-vous supprimez a partire du numero ...(4)"<<endl;
cin>>choix1;
            switch(choix1)
            {
                case 3:
                {
                    cout<<"Veuillez saisir le nom :"; /*demander a lutilisateur de saisir le nom
qu il veux supprimer */
                    cin>>nom; // lire la reponse
                    for(i=0;i<=5;i++)
                    {
                        if(nom==Tnom[i])
                        {
                            cout<<" le nom est :"<<Tnom[i]; //afficher le nom
                            Tnom[i]=' '; //ecrasé le contenu par un espace
                            cout<<" Le num de place est :"<<Tnum[i]; // afficher le numero de place
                            Tnum[i]=0; //ecrasé le contenu par un zero
                        }
                    }
                }
                case 4:
                {
                    if(choix1==4)
                    {
                        cout<<"Veuillez saisir le numero :"; /*demander a lutilisateur de saisir le
numero de place qu il veux supprimer*/
                        cin>>place; // lire la reponse
                        for(i=0;i<=5;i++)
                        {
                            if(place==Tnum[i])
                            {
                                cout<<" le nom est :"<<Tnom[i]; //afficher le nom
                                Tnom[i]=' '; //ecrasé le contenu par un espace
                                cout<<" Le num de place est :"<<Tnum[i]; // afficher le numero de place
                                Tnum[i]=0; //ecrasé le contenu par un zero
                            }
                        }
                    }
                }
            }
            case 3:
            default;; // pour exite le menu
        }
    }
    cout<<"voulez vous recommencez?(oui/non)"<<endl; /*demander a lutilisateur s'il veux
continuer */
    cin>>rep; //lire la reponse
}while(rep=="oui");
}
```

Exercice 21 :

/* Le tri par sélection :

```
On cherche le plus petit élément du tableau et on le place en 1er, puis on cherche le plus petit dans ce qui reste et on le met en second, etc */
#include <cstdlib> /* On inclure la bibliothèque qui sert principalement à déclarer la fonction C system()*/
#include <iostream> /****** On inclure la bibliothèque des entrées/sorties *****/
#include <string> //on inclure la classe string pour travailler sur les chaînes de caractères
using namespace std; //espace de nom
int main(int argc, char *argv[]) // le main
{
    int i,j,k,min,pos; //déclaration des variables de type entier
    int tab[10]={52,10,1,25,62,3,8,55,3,23}; //déclaration d'un tableau de type entier
    for(i=0;i<9;i++){ //une boucle
        pos=i; // On initialise la position de i
        min=tab[i];
        for(j=i+1;j<10;j++){//une boucle
            if(tab[j]<min){ //On cherche le min dans le tableau
                min=tab[j];
                pos=j; //On mémorise sa position
            }
        }
        for(k=pos;k>=i+1;k--){ //Une boucle
            tab[k]=tab[k-1]; //le décalage jusqu'à la position i+1
        }
        tab[i]=min;
    }
    cout<<"Le tableau trié est le suivant: "<<endl;
    for(i=0;i<10;i++){ //On affiche le tableau trié
        cout<<"tab["<<i<<"]="<<tab[i]<<endl;
    }
}
system("PAUSE");
}
```

Exercice 22 :

/* Le tri par bulle :

Un tri bulle est un tri plus astucieux, son principe est de faire remonter petit à petit un élément trop grand vers le haut du tableau en comparant les éléments deux à deux. Si l'élément de gauche est supérieur à son voisin de droite on les inverse et on continue avec le suivant. Lorsque l'on est en haut du tableau on repart au début et on s'arrête lorsque tous les éléments sont bien placés.

On a parcouru tout le tableau, on recommence, jusqu'à ce que tout soit bien placé.*/

```
#include <cstdlib> /*On inclure la bibliothèque qui sert principalement à déclarer la fonction C system()*/
#include <iostream> /****** On inclure la bibliothèque des entrées/sorties *****/
#include <string> //on inclure la classe string pour travailler sur les chaînes de caractères
using namespace std; //espace de nom
int main(int argc, char *argv[]) // le main
{
    int i,j,nb,tmp; //déclaration des variables de type entier
    int tab[10]={52,10,1,25,62,3,8,55,3,23}; //déclaration d'un tableau de type entier
    bool trié; // déclaration des variables de type booléenne
    j=9; //On initialise le j par 9
    do
    {
        trié=false; //On initialise la variable trié par false
        for(i=0;i<=j;i++){ //Une boucle
```

```

        if(a[i]>a[i+1]) //On compare l'elemnt du tableau avec son voisin
        {
            tmp=a[i+1];
            a[i+1]=a[i]; // on purmute les element //
            a[i]=tmp; // du tableau //
            trie=true; ///////////////////////////////////////////////////////////////////
        }
    }
    j--;
}while(j!=0 && trie==true);
cout<<"VOICI LE TABLEAU TRIE :"<<endl;
for(i=0;i<10;i++) //Affichage du tableau trier
    cout<<"a["<<i<<"]="<<a[i]<<endl;
}

```

Exercice 23 :

```

#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/
sorties *****/
#include <string> //on inclure la classe string pour travailler sur les chaine de caractere
using namespace std; //espace de nom
int main(int argc, char *argv[]) // le main
{
    int i,j,k,permut; //declaration des variables de type entier
    int tab[10]={52,10,1,25,62,3,8,55,3,23}; //declaration d'un tableaux de type entier

    for(i=0;i<=8;i++){ //Une boucle
        if(tab[i]>tab[i+1]){ //On compare l'elemnt du tableau avec son voisin
            permut=tab[i+1]; //On stock la valeur dans la variable permut
            k=0;
            while(permut>=tab[k]){
                k++;
            }
            for(j=i+1;j>k;j--){ //Une boucle pour le decalage
                tab[j]=tab[j-1];
            }
            tab[k]=permut;
        }
    }
    cout<<"VOICI LE TABLEAU TRIE :"<<endl;
    for(i=0;i<10;i++) //Affichage du tableau trier
        cout<<"a["<<i<<"]="<<a[i]<<endl;
    system("pause");
}

```

Exercice 25:

```

/*
*****
*****
*****
*****
*****
****
***
**
* */

#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;

```

```
int main(int argc, char *argv[])
{
    int i,k;
    for(k=14;k>=0;k--)
    {
        for(i=0;i<=k;i++)
        {
            cout<<"*";

        }
        k--;

    }
    cout<<endl;

}
system("pause");

}
```

Exercice 26 :

```
/*
    *
    **
    ***
    ****
    *****
    *
*/
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;

int main(int argc, char *argv[])
{

    int i,k,m;

    m=15;

    for(k=0;k<=14;k++)
    {
        cout<<setw(m);
        for(i=k;i>=0;i--)
        {

            cout<<"*";

        }

        m--;
        cout<<endl;

    }

}
```

```
    system("pause");  
}
```

Exercice 27:

```
/*  
  
 *  
 **  
 ***  
 ****  
 *****  
 ****  
 ***  
 **  
 *  
 */  
#include <cstdlib>  
#include <iostream>  
#include <iomanip>  
using namespace std;  
  
int main(int argc, char *argv[])  
{  
    int i, j;  
  
    for(i=0; i<=14;i++)  
    {  
        for(j=0; j<=i;j++)  
        {  
            cout<<"*";  
        }  
        cout<<endl;  
    }  
  
    system("PAUSE");  
}
```

Exercice 28:

```
/*  
  
 *  
 ***  
 *****  
 ****  
 ***  
 **  
 *  
 */  
#include <cstdlib>  
#include <iostream>  
#include <iomanip>  
using namespace std;  
  
int main(int argc, char *argv[])
```

```
{
  int i, j, m;

  m=7;

  for(i=0; i<=13;i+=2)
  {
    cout<<setw(m);
    for(j=0; j<=i;j++)
    {
      cout<<"*";
    }
    cout<<endl;

    m--;
  }

  system("PAUSE");
}
```

Exercice 29:

```
/*
*****
*****
*****
*****
*****
***
*
*/

#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;

int main(int argc, char *argv[])
{
    int i, k, m;
    m=0;
    for(k=11; k>=0; k-=2)
    {
        m++;
        cout<<setw(m);
        for(i=0; i<=k; i++)
        {
            cout<<"*";
        }
        cout<<endl;
    }
    system("pause");
}
```


Exercice 31

Version 1:

```
#include <cstdlib>
#include <iostream>
#include <string>
using namespace std;

int main(int argc, char *argv[])
{
    /* declaration des variable */
    string rep;
    int somme;

    do
    {
        system("cls");//vider l ecran

        cout<<"veuillez SVP entrez une somme d'argent";
        cin>>somme;

        system("cls");
        cout<<"la decomposante du "<<somme<<" Dhs est :"<<endl<<endl;

        do
        {
            int nbr =0;

            while(somme>=200)
            {
                somme =somme-200;
                nbr++; //on compte le nombre de billet de 200 DH
            }

            if(nbr!=0)
                cout<<nbr<<" billets de 200 Dhs"<<endl;

            if(somme>=100)
            {
                somme = somme-100;
                cout<<"1 billet de 100 Dhs"<<endl;
            }
            else
            {
                if( somme>=50)
                {
                    somme =somme-50;
                    cout<<"1 biellet de 50 Dhs"<<endl;
                }

                nbr =0;
                while(somme>20)
                {
```

Programmation structurée

```
        somme =somme-20; //on compte le nombre de billet de 20 DH
        nbr++;
    }
    if(nbr!=0)
    cout<<nbr<<" billets de 20 Dhs"<<endl;

    if(somme>=10)
    {
        somme =somme-10;
        cout<<"1 piece de 10 Dhs"<<endl;
    }
    else
    {
        if( somme>=5)
        {
            somme =somme-5;
            cout<<"1 piece de 5 Dhs"<<endl;
        }
        else
        {
            nbr =0;
            while(somme>=2)
            {
                somme =somme-2; //on compte le nombre de piece de 200 DH

                nbr++;
            }
            if(nbr!=0)
            cout<<nbr<<" pieces de 2 Dhs"<<endl;

            if( somme>=1)
            {
                somme=somme-1;
                cout<<"1 piece de 1 Dhs"<<endl;
            }
        }
    }
}

}while(somme!=0); /*tant que la somme est differente a 0 on reboucle jusqu'a
la somme est = a 0*/

    cout<<"Voulez-vous continuer ? (oui/non)";
    cin>>rep;
}while(rep=="oui");

    system("PAUSE");
}
```

Version 2:

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int tab[8]={ 200,100,50,20,10,5,2,1 }; // initialisation et declaration du tableau
    int somme,z , i =0;

    cout<<"Veuillez entrer une somme d'argents :";
    cin>>somme;
    do
    {
        if(somme>=tab[i])
        {
            z=somme/tab[i];
            somme=somme-(tab[i]*z);
            if(i<4)
                cout<<z<<" billet(s) de "<<tab[i]<<endl;
            else
                cout<<z<<" piece(s) de "<<tab[i]<<endl;
        }

        i++;

    }while(somme !=0);

    system("PAUSE");
}
```

Exercice 32:

```
#include <iostream>
using namespace std;
const int N=20;

int main(){
    int n,M,i,j,tab[N];
    cout<<"Veuillez saisir la taille du tableau n= ";
    cin>>n;
    for(i=0;i<n;i++){ //une boucle pour remplir le tableau
        cout<<"tab["<<i<<"]=" ";
        cin>>tab[i];
    }
    cout<<"Veuillez SVP saisir un entier M= ";
    cin>>M;
    j=0;
    for(i=0;i<n;i++) //on cherche toutes les occurance de l'entier saisie qui existe dans le
tableau
        if(tab[i]==M){
            j++;
            cout<<"l'occence numeros "<<j<<" se trouve dans l'indice "<<i<<endl;
        }
    system("pause");
    return 0;
}
```

```
}
```

Exercice 33:

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int n, x,S1,S2,c;
    S1=0;
    // on saisie notre entier
    cout<<"veuillez saisir un entier SVP ";
    cin>>n;

    do
    {
        x= n%2;
        if(x==0) //si x est paire alors
        {
            n=n/2;
        }
        Else // si x est impaire alors
        {
            n=(n*3)+1;
        }
        S1=S1+1;
        cout<<n;
        if(n>1)
            cout<<" ";
    }while(n>1);

    cout<<endl<<"Le nombre de fois est : "<<S1<<endl; /*on affiche le nombre de
transformation faite */

    system("PAUSE");
}

```

Exercice 34 :

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int L,C,i,j;
    L=3;
    C=2;
    int M[L][C];
}

```

```
int V[L*C];
//ces deux boucle sont utilisées pour remplir la matrice

for(i=1;i<=L;i++) //une boucle pour remplir les lignes
{
    for(j=1;j<=C;j++) //une boucle pour remplir les colonnes
    {
        cout<<"element ("<<i<<","<<j<<") :";
        cin>>M[i][j];
    }
}
int k=0;
// on transfere la matrice dans un tableau d'un seul dimension
for(i=1;i<=L;i++)
{
    for(j=1;j<=C;j++)
    {
        V[k]=M[i][j]; //
        k++;
    }
}
system("cls");
cout<<"voici le nouveau tableau"<<endl;

for(i=0;i<=k-1;i++)
{
    cout<<V[i]<<endl; //affichage du tableau
}

system("PAUSE");

}
```

Exercice 35 :

```
#include <cstdlib>
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
{
    int Annee,salaire,age,Rsalaire;
    // on saisie les les information concernant le cadre
    cout<<"Années d'enciente ";
    cin>>Annee;
    cout<<"Votre salaire ";
    cin>>salaire;
    cout<<"Votre age ";
    cin>>age;

    Rsalaire = (salaire/2)*Annee;

    if(age>45 && age<49) // si l'age du cadre est comprise entre 45 et 49 ans alors :
```

```
{
    salaire = Rsalaire +(salaire*2);
}
else
{
    if(age>50) // si l'age du cadre est superieur à 50 ans
        salaire = Rsalaire +(salaire*5);
}

cout<<endl<<endl<<"votre indemnite est :"<<Rsalaire<<" Dhs"<<endl;
system("PAUSE");
}
```

Exercice 36:

```
#include <cstdlib>
#include <iostream>
```

```
using namespace std;
```

```
int main()
{
    int n;
    // on saisie l'ordre de la matrice
    cout<<"veuillez saisir les lignes et les colone"<<endl;
    cin>>n;
    int a[n][n],Ta[n][n]; // declaration de deux tableau (n case)
    // Le remplissage de la matrice
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            cout<<"veuillez svp saisir un entier de la case ("<<i<<","<<j<<") du
matrice"<<endl;
            cin>>a[i][j];
        }
    }
    // on affiche la matrice comme des lignes et des colones
    cout<<"voila votre Matrice saisie"<<endl;
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            cout<<" "<<a[i][j]<<" ";
        }
        cout<<endl;
    }
    // on fait une transformation cette matrice
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            // Les lignes deviennent des colones et les colones deviennent des lignes
            Ta[j][i]=a[i][j];
        }
    }
}
```

```
    }  
    // on affiche notre matrice apres la transformation  
    cout<<"votre Matrice est devenu"<<endl;  
    for(int i=0;i<n;i++)  
    {  
        for(int j=0;j<n;j++)  
        {  
            cout<<" "<<Ta[i][j]<<" ";  
        }  
        cout<<endl;  
    }  
  
    system("PAUSE");  
}
```

Exercice 37 :

Version 1 :

```
#include <cstdlib>  
#include <iostream>  
#include <iomanip>  
using namespace std;
```

```
int main(int argc, char *argv[])  
{
```

```
    int P[3][3];  
    int i,j;
```

```
    for(i=1;i<=3;i++) //la boucle pour remplir les lignes de la matrice  
    {  
        for(j=1;j<=3;j++) //la boucle pour remplir les colonnes de la matrice  
        {  
            cout<<"element ("<<i<<","<<j<<") :";  
            cin>>P[i][j];  
        }  
    }
```

```
    int somme =0 ;  
    system("cls");  
    cout<<"les elements sur la diagonale son :"<<endl;  
    for (i=1; i<=3; i++)  
    {  
        for (j=1; j<=3; j++) //on parcour la matrice pour trouver la somme de la diagonale  
        {  
            if(i==j)  
            {  
                cout<<P[i][i]<<endl; //pour terminer avec 1  
                somme=somme+P[i][i];  
            }  
        }  
    }
```

```
    cout<<endl<<"la somme des elements sur la diagonale son :"<<somme<<endl;
```

```
system("PAUSE");
```

```
}
```

Version 2:

```
#include <cstdlib>
#include <iostream>
#include <iomanip>

using namespace std;

int main(int argc, char *argv[])
{
    int res,i,j,n;
    // on saisie l'ordre de la matrice
    cout<<"veuillez saisir le nombre de matrice"<<endl;
    cin>>n;
    int t[n][n];
    // Le remplissage de la matrice
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            cout<<"veuillez svp saisir un entier de la case ("<<i<<","<<j<<") du
matrice"<<endl;
            cin>>t[i][j];
        }
    }

    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            cout<<" "<<t[i][j]<<" ";
        }
        cout<<endl;
    }

    res=0;
    // on calcule la somme des element de la diagonale
    cout<<"le diagonale est : "<<endl;
    for(i=0;i<n;i++)
    {
        if(i==0)
        {
            cout<<setw(i*3)<<" "<<t[i][i]<<" "<<endl;
            res=res+t[i][i];
        }
        else
        {
            res=res+t[i][i];
            cout<<setw(i*3+1)<<" "<<t[i][i]<<" "<<endl;
        }
    }

    cout<<"Le resultat de la trace est : "<<res<<endl;
    system("PAUSE");
}
```

Exercice 48:

```
#include <iostream>
using namespace std;
const int N=10;
int main(){
    int tab[11];
    int n,x,i,j;
    // une boucle pour remplir le tableau
    for(i=0;i<10;i++){
        cout<<"tab["<<i<<"]=" ";
        cin>>tab[i];
    }
    cout<<"Veuillez saisir un entier X= ";
    cin>>x;
    i=-1;
    // on cherche la position où on doit mettre notre X
    do{
        i++;
    }while(x>tab[i] && i!=10);
    if(i==10){
        tab[10]=x;
    }
    else{
        // on fait decaler les element du tableau jusqu'à la position i trouver
        for(j=10;j>i;j--){
            tab[j]=tab[j-1];
        }
        tab[i]=x;
    }
    // on affiche notre tableau apres l'operation
    for(i=0;i<11;i++){
        cout<<"tab["<<i<<"]=" "<<tab[i]<<endl;
    }
    system("pause");
    return 0;
}
```

Exercice 49:

```
#include <iostream>
using namespace std;
const int N=10;

int main()
{
    int tab[N],n,x, i,j=0;
    cout<<"Veuillez saisir la taille du tableau N= ";
    cin>>n;
    // on remplit notre tableau
    for (i=0; i<n; i++)
    {
        cout << "tab[" << i <<"]=" ";
        cin >> tab[i];
    }
    // on saisie un entier X
    cout<<"Veuillez saisir un entier X= ";
```

```
cin>>x;
// on elimine tous les occurrence de X dans le tableau
for(i=0; i<n; i++){
    if(tab[i]!=x) {
        tab[j]=tab[i];
        j++;
    }
}
// on affiche que les element qui sont differents de X

for(i=0; i<j; i++){
    cout<<"Tab["<<i<<"]="<<tab[i]<<endl;
}
system("pause");
return 0;
}
```

Exercice 40:

```
#include <iostream>
#include <iomanip>
using namespace std;
int main(){
    int op,n,i,k,pos,j;
    k=0;
    for(i=1;i<10;i++){
        k=(k*10)+i;
        op=(k*8)+i;
        cout<<setw(20)<<k<<" * 8"<<" +1= "<<op<<endl;
    }
    system("pause");
    return 0;
}
```

Exercice 41:

```
#include <cstdlib>
#include <iostream>
#include <string>
using namespace std;

int main(int argc, char *argv[])
{
    int i,j; // declaration des variables de type entier
    char tab[50]={' '}; // declaration d un tableau char et initialiser par des espaces

    cout<<"Veuillez saisir une phrase qui termine par un '.'<<endl;
    cin.getline(tab,50);

    j=0; // initialisation par 0
    i=0; // initialisation par 0

    do
    {
        cout<<"Mot "<<j+1<<" : "; // au moins on affiche un mot
        do
        {
```

```
/* si la valeur de la case du tableau est different a espace on affiche cette valeur , et aussi
different un '.' ( terminateur)*/
if(tab[i]!=' ' && tab[i]!='.')
{
    cout<<tab[i]; //affichage des caracteres
}
i++; // incrementation du compteur
}while( tab[i-1]!=' ' && tab[i-1]!='.');// la condition dans while pour sortire de la boucle

j++; // incrementation du compteur
cout<<endl; // apres chaque mot , on est besoin de retourner a la ligne
}while( i!= 50 && tab[i-1]!='.');// la condition dans while pour sortire de la boucle

system("PAUSE");// bloquer l ecran

}
```

Exercice 44:

```
#include <cstdlib>
#include <iostream>
using namespace std;
// Cette fonction cherche le nombre de valeurs comprises entre 0 et 10
int F(int t1[ ],int n)
{
    int nb,i,j;
    int t2[n];

    nb=0;
    j=0;
    cout<<endl<<endl<<"les valeurs sont : "<<endl;
    for(i=0;i<n;i++)
    {
        if(t1[i]>0 && t1[i]<10)
        {
            nb=nb+1;
            cout<<t1[i]<<" / ";
            t2[j]=t1[i];
            j++;
        }
    }
    return nb; // le nombre de valeur comprise entre 0 et 10
}

int main(int argc, char *argv[])
{
    int n=10;
    int tab[n];
    // Le remplissage du tableau
    for(int i=0;i<=9;i++)
    {
        cout<<"Element "<<i+1<<" : ";
        cin>>tab[i];
    }
}
```

```
    }
    system("cls");
    cout<<" le tableau d'origine est : "<<endl<<endl;
    for(int i=0;i<=9;i++)
        cout<<tab[i]<<" | ";
    // on renvoie les resultat calculer dans la fonction
    int X = F(tab,n);
    cout<<endl<<endl ;
    cout<<"le nombre de valeurs comprises entre 0 et 10 dans le tableau :
"<<endl<<X<<endl;
    system("PAUSE");
```

```
}
```

Exercice 43:

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    int i,j;
```

```
    // Le remplissage de la matrice
```

```
    int Tableau[3][4];
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<4;j++)
```

```
        {
```

```
            cout<<"element ("<<i<<","<<j<<") :";
```

```
            cin>>Tableau[i][j];
```

```
        }
```

```
    }
```

```
}
```

```
// on initialise le min et le max par le 1er element de la matrice
```

```
int min = Tableau[0][0];
```

```
int max =Tableau[0][0];
```

```
// on cherche le min et le max dans notre matrice
```

```
for(i=0;i<3;i++)
```

```
{
```

```
for(j=0;j<4;j++)
```

```
{
```

```
if(Tableau[i][j]<min)
```

```
{
```

```
min=Tableau[i][j];
```

```
}
```

```
else
```

```
{
```

```
if(Tableau[i][j]>max)
```

```
{
```

```
max =Tableau[i][j];
```

```
}
```

```
}
```

```
    }
}
}
// on affiche le min et le max trouver
cout<<"le minimum des entiers est : "<<min<<endl;
cout<<"le maximum des entiers est : "<<max<<endl;
system("PAUSE");
}
```

Exercice 44: Version 1 :

```
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <conio.h>
using namespace std;

int main(int argc, char *argv[])
{
    int multi ,i,j,m;
    /* la boucle de l affichage horizontale*/
    cout<<"X*Y ";
    for(j=0;j<=10;j++)
        cout<<j<<setw(5)<<" ";
    cout<<endl;
    cout<<"-----" <<endl;
    /*la boucle de l affichage vertical*/
    for(j=0;j<=10;j++)
    {
        /* decalage pour le dernier ligne*/
        if(j!=10)
            cout<<" "<<j;
        else
            cout<<j;

        /*la multiplication */
        for(i=0; i<=10; i++)//boucle pour
        {

            cout<<setw(5)<<j*i<<" ";
            //affichage de la table de multiplication
        }

        cout<<endl;
    }

    system("PAUSE");
}
```

Version 2 :

```
#include <iostream>
#include <math.h>
#include <iomanip>
```

```
using namespace std;
const int N=20;

int main(){
    int n,i,j,k;
    //cout<<"Veuillez saisir la taille du tableau n= ";
    //cin>>n;
    cout<<"le plroduit scalaire est: "<<endl;
    bool test;
    cout<<"X*Y";
    cout<<setw(5)<<"0";
    /* la boucle de l affichage horizontale*/

    for(i=1;i<=10;i++){
        cout<<setw(7)<<i;
        }
        cout<<endl;
    for(i=0;i<78;i++){
        cout<<"-";
        }
    cout<<endl;

    /*la boucle de l affichage vertical*/
    for(int j=0;j<=9;j++){
        cout<<j;
        test=false;
    //affichage de la tabelle de multiplication

        for(i=0;i<=10;i++){
            if((i*j)>=10 && test==false)
            {
                cout<<setw(8)<<j*i;
                test=true;
            }
            else
                cout<<setw(7)<<j*i;

            }

        cout<<endl;
        }
        cout<<"10";
        cout<<setw(6)<<"0";
        cout<<setw(8)<<"10";
        for(k=2;k<=10;k++){
            cout<<setw(7)<<10*k;

            }
        cout<<endl;

    cout<<endl;
    system("pause");
    return 0;
}
```

Exercice 45 :

```
#include <iostream>
#include <math.h>
using namespace std;
const int N=20;

int main(){
    int n,i,j; /*declaration des variables */

    float tab1[N],tab2[N],k; /*declation des tableaux */

    cout<<"Veuillez saisir la taille du tableau n= ";
    cin>>n;
    /*Le remplissage du 1er tableau */

    for(i=0;i<n;i++){
        cout<<"tab1["<<i<<"]=" ";
        cin>>tab1[i];
    }
    /*Le remplissage du 2eme tableau */

    for(i=0;i<n;i++){
        cout<<"tab2["<<i<<"]=" ";
        cin>>tab2[i];
    }

    k=0;
    /*calcul du produit scalaire */

    for(i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if(i==j){
                k=k+(tab1[i]*tab2[j]);
            }
        }
    }
    /*Affichage du resultat */

    system("cls");
    cout<<"le plroduit scalaire est: "<<endl;
    cout<<"( ";
    for(i=0;i<=2;i++){
        cout<<tab1[i]<<" ";
        cout<<") * ( ";
    }
    for(i=0;i<=2;i++){
        cout<<tab2[i]<<" ";
        cout<<") =";
    }
    for(i=0;i<2;i++){
        cout<<"( ";
        cout<<tab1[i]<<" ";
        cout<<") * ( ";
        cout<<tab2[i]<<" ";
        cout<<") + ";
    }

    cout<<"( ";
    cout<<tab1[2]<<" ";
```

```
cout<<" * (" ;
cout<<tab2[2]<<" ";
cout<<"= "<<k<<endl;
system("pause");
return 0;
}
```

Exercice 46 :

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])

{
/* Déclarations des tableaux et leurs dimensions */
int A[50], B[50], FUS[100];
int N, M;
int iA, iB, iFUS;

/* Saisie des données du tableau A */
cout<<"Dimension du tableau A : ";
cin>>N;
cout<<"Entrer les éléments de A dans l'ordre croissant : "<<endl;
for (iA=0; iA<N; iA++)
{
    cout<<"Elément A["<<iA+1<<"] : ";
    cin>>A[iA];
}

/* Saisie des données du tableau B*/
cout<<"Dimension du tableau B";
cin>>M ;

cout<<"Entrer les éléments de B dans l'ordre croissant : "<<endl;
for (iB=0; iB<M; iB++)
{
    cout<<"Elément B["<<iB+1<<"] : ";
    cin>>B[iB];
}

/* Fusion des éléments de A et B dans FUS */

iA=0;
iB=0;
iFUS=0;

while (iA<N && iB<M)
    if(A[iA]< B[iB])
    {
        FUS[iFUS]=A[iA];
        iFUS++;
        iA++;
    }
}
```

```
        }
        else
        {
            FUS[iFUS]=B[iB];
            iFUS++;
            iB++;
        }

/* pour copier le reste du tableau A si il n'est pas encor a la fin */
while (iA<N)
{
    FUS[iFUS]=A[iA];
    iFUS++;
    iA++;
}
/* pour copier le reste du tableau B si il n'est pas encor a la fin */
while (iB<M)
{
    FUS[iFUS]=B[iB];
    iFUS++;
    iB++;
}
cout<<endl<<endl;
cout<<"Tableau FUS trie par ordre croissant : " <<endl;
for (iFUS=0; iFUS<N+M; iFUS++)
{
    cout<<FUS[iFUS];
    if(iFUS<(N+M)-1)
        cout<<"/";
}
cout<<endl;
system("PAUSE");
}
}
```

Exercice 47 :

Version 1 :

```
#include <cstdlib>
#include <iostream>
#include <stdio.h>
#include <iomanip>
using namespace std;

int main()
{
    /* Déclarations */
    int P[20][20];
    int N;
    int i,j;
    /* Saisie des données */
    do {
        cout<<"veuillez entrez le degre N du triangle ( entre 1 et 19) : ";
        cin>>N;
```

```
} while (N>19|| N<0);

for (i=0; i<=N; i++)
{
    P[i][0]=1; //pour commencé avc 1
    for (j=1; j<i; j++)
        P[i][j] = P[i-1][j] + P[i-1][j-1]; //pour faire 1 addition

    P[i][i]=1; //pour terminer avc 1
}
/* Affichage du résultat */

system("cls");

cout<<"Triangle de Pascal de degre " <<N<<endl<<endl;

for (i=0; i<=N; i++)
{
    for (j=0; j<=i; j++)
        cout<<setw(5)<< P[i][j];
    cout<<endl;
}

system("PAUSE");

}
```

Version 2 :

```
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;

int main(int argc, char *argv[])
{
    int i,j,X,n;
    cout<<"Veuillez entrer un entier ";
    cin>>n;
    cout<<setw(5);

    for(i=0;i<n;i++)
    {
        X=1;
        for(j=i;j>=0;j--)
        {

            /* Affichage du résultat */
            cout<<X<<setw(5);
            X=(X*j)/((i-j)+1);

        }
    }
}
```

```
    cout<<endl;
}

system("PAUSE");
}
```

Exercice 48 :

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int L,C,i,j;
    L=3;
    C=2;
    int M[L][C];
    int V[L*C];
    //ces deux boucle sont utilisées pour remplir la matrice

    for(i=1;i<=L;i++) //une boucle pour remplir les lignes
    {
        for(j=1;j<=C;j++) //une boucle pour remplir les colonnes
        {
            cout<<"element ("<<i<<","<<j<<") :";
            cin>>M[i][j];
        }
    }
    int k=0;
    // on transfere la matrice dans un tableau d'un seul dimension
    for(i=1;i<=L;i++)
    {
        for(j=1;j<=C;j++)
        {
            V[k]=M[i][j]; // on met chaque element de la matrice dans le tableau
            k++;
        }
    }
    system("cls");
    cout<<"voici le nouveau tableau"<<endl;

    for(i=0;i<=k-1;i++)
    {
        cout<<V[i]<<endl; //affichage du tableau
    }

    system("PAUSE");
}
```

Exercice 49 :

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main()
{
    int N,M;
    // on saisie l'ordre de la matrice
    cout<<"veuillez saisir la logueur des lignes "<<endl;
    cin>>N;
    cout<<"veuillez saisir la longueur des colones"<<endl;
    cin>>M;

    int a[N][M],Ta[M][N]; // declaration de deux tableau (n case)
    // Le remplissage de la matrice
    for(int i=0;i<N;i++)
    {
        for(int j=0;j<M;j++)
        {
            cout<<"veuillez svp saisir un entier de la case ("<<i<<","<<j<<") du
matrice"<<endl;
            cin>>a[i][j];
        }
    }
    // on affiche la matrice comme des lignes et des colones
    cout<<"voila votre Matrice saisie"<<endl;
    for(int i=0;i<N;i++)
    {
        for(int j=0;j<M;j++)
        {
            cout<<" "<<a[i][j]<<" ";
        }
        cout<<endl;
    }
    // on fait une transformation cette matrice
    for(int i=0;i<N;i++)
    {
        for(int j=0;j<M;j++)
        {
            // Les lignes deviennent des colones et les colones deviennent des lignes
            Ta[j][i]=a[i][j];
        }
    }
    // on affiche notre matrice apres la transformation
    cout<<"votre Matrice est devenu"<<endl;
    for(int i=0;i<M;i++)
    {
        for(int j=0;j<N;j++)
        {
            cout<<" "<<Ta[i][j]<<" ";
        }
        cout<<endl;
    }
}
```

```
system("PAUSE");
```

```
}
```

Exercice 50 :

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    int i,j;
```

```
    char M[3][4]={'a','b','c','d','e','f','g','h','i','j','k','l'};
```

```
    char V[3][4];
```

```
    int K=0;
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<4;j++)
```

```
        {
```

```
            V[i][j]=M[i][j];
```

```
        }
```

```
        cout<<endl;
```

```
    }
```

```
    cout<<" /    \\"<<endl;
```

```
        for(i=0;i<3;i++)
```

```
    { cout<<"|";
```

```
        for(j=0;j<=3;j++)
```

```
        {
```

```
            cout<<V[i][j]<<" ";
```

```
        }
```

```
        cout<<"|"<<endl;
```

```
    }
```

```
    cout<<" \\\    /"<<endl<<endl;
```

```
////////////////////////////////////
```

```
    cout<<" /    \\"<<endl;
```

```
        for(i=0;i<3;i++)
```

```
    { cout<<"|";
```

```
        for(j=0;j<=3;j++)
```

```
        {
```

```
            cout<<V[i][j]<<" ";
```

```
        }
```

```
    }
```

```
    cout<<"|"<<endl;
```

```
}
```

```
cout<<" \\  
  cout<<endl;  
cout<<endl;  
cout<<endl;  
////////////////////  
  
cout<<" /          \\  
for(i=0;i<3;i++)  \\  
{ cout<<"|";  
  for(j=0;j<=3;j++)  
  {  
    cout<<V[i][j]<<" +";  
    cout<<V[i][j]<<" ";  
  }  
  
  cout<<"|<<endl;  
}  
cout<<" \\  
  cout<<endl; cout<<endl; cout<<endl;  
  
system("PAUSE");  
  
}
```

Exercice 51 :

```
#include <cstdlib>  
#include <iostream>  
using namespace std;  
  
int main(int argc, char *argv[])  
{  
  
  int tab[3];  
  int tabl[3];  
  int multi1,multi,i;  
  // Le remplissage du 1er tableau  
  
  for(i=0;i<=2;i++)  
  {  
    cout<<"Element "<<i+1<<" : ";  
    cin>>tab[i];  
  }  
  // Le remplissage du 2ème tableau  
  for(i=0;i<=2;i++)  
  {  
    cout<<"Element "<<i+1<<" : ";  
    cin>>tabl[i];  
  }  
  
  multi=0;  
  /* On multiplie chaque élément du tableau 1 par chaque élément du tableau 2, et  
  additionner le tout.*/  
  
  for(i=0;i<=2;i++)  
  { for(int j=0;j<=2;j++){
```

```
    multi=multi+(tab[i]*tabl[j]);
    }
    }
    // On affiche le résultat
    cout<<"l'addition de tout les multiplication des schtroupmf : "<<multi<<endl;
    system("PAUSE");
}
}
```

Exercice 52 :

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int tab[10]; // La declaration du tableau
    int i,x,k,j; // La declaration des variable
    cout<<"veuillez saisir votre tableau: "<<endl;
    // Le remplissage du tableau
    for(i=0;i<10;i++)
    {
        cout<<"tab["<<i<<"]=" ";
        cin>>tab[i];
    }
    x=-1;

    do{
        x++; // pour afficher les nombre de 1 à 20
        k=0;
        for(j=0;j<10;j++)
        {
            if(tab[j]==x){
                k++; // pour compter combien de fois on a nombre
            }
        }
    }
    // Affichage du resultat
    cout<<"Le nombre de "<<x<<" dans le tableau est = "<<k<<" fois"<<endl<<endl;

    }while(x!=20);
    system("pause");
    return 0;
}
```

Exercice 53 :

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
```

```
{
  /* Déclarations des tableaux et leurs dimensions */
  int T[50], V[50], R[100];
  int N, M;
  int iT, iV, iR;

  /* Saisie des données du tableau A */
  cout<<"Dimension du tableau A : ";
  cin>>N;
  cout<<"Entrer les éléments de A dans l'ordre croissant : "<<endl;
  for (iT=0; iT<N; iT++)
  {
    cout<<"Élément A["<<iT+1<<"] : ";
    cin>>T[iT];
  }

  /* Saisie des données du tableau B */
  cout<<"Dimension du tableau B";
  cin>>M;

  cout<<"Entrer les éléments de B dans l'ordre croissant : "<<endl;
  for (iV=0; iV<M; iV++)
  {
    cout<<"Élément B["<<iV+1<<"] : ";
    cin>>V[iV];
  }

  /* Fusion des éléments de A et B dans FUS */

  iT=0;
  iV=0;
  iR=0;

  while (iT<N && iV<M)
  if(T[iT]< V[iV])
  {
    R[iR]=T[iT];
    iR++;
    iT++;
  }
  else
  {
    R[iR]=V[iV];
    iR++;
    iV++;
  }

  /* pour copier le reste du tableau T si il n'est pas encor a la fin */
  while (iT<N)
  {
    R[iR]=T[iT];
    iR++;
    iT++;
  }

  /* pour copier le reste du tableau V si il n'est pas encor a la fin */
  while (iV<M)
```

```
{
    R[iR]=V[iV];
    iR++;
    iV++;
}

cout<<endl<<endl;

cout<<"Tableau FUS trie par ordre croissant : " <<endl;
for (iR=0; iR<N+M; iR++)
{
    cout<<R[iR];
    if(iR<(N+M)-1)
        cout<<" ";
}
system("PAUSE");
}
```

Exercice 54 :

```
#include <cstdlib>
#include <iostream>
#include <conio.h>
#include <stdlib.h>
#include <iomanip>
#include <string>
using namespace std;

/***** prototype des fonctions *****/
int mprincipale();
void afficher (int affiche[],string message);
void remplir (int tremplir[]);
int emin (int temin[]);
int emax ( int temax[]);
void triselection ( int tselection[]);
void tribulle(int tbulle[]);
void tripermutation (int tpermutation[]);
void tricomptage (int tcomptage[]);
void rechercher(int trech[],int X);

/***** Le Main *****/
int main()
{
    int tvaleur[10]={0,0,0,0,0,0,0,0,0,0};
    string rep; // par la rep 'oui' ou 'non'
    int choice,Xmin,Xmax;
    do
    {
        choice = mprincipale(); //1 appel de la fonction mprincipale qui donne le menu
        principale

        switch (choice) // selon le choix on vas choisir un un procedur .
```

```
{
case 1:
    remplir(tvaleur); // l'appel de la procedur remplir.
    cout<<endl<<endl;
    system("pause");
    break;
case 2:
    Xmin=emin(tvaleur); // l'appel de la procedur emin.
    cout<<endl<<"le min est : "<<Xmin<<endl;
    cout<<endl<<endl;
    system("pause");
    break;
case 3:
    Xmax = emax(tvaleur); // l'appel de la procedur emax.
    cout<<endl<<" le max esr : "<<Xmax<<endl;
    cout<<endl<<endl;
    system("pause");
    break;
case 4:
    triselection(tvaleur); // l'appel de la procedur qui fait le trie de selection.
    cout<<endl<<endl;
    system("pause");
    break;
case 5:
    tribulle(tvaleur); // l'appel de la procedur qui fait le trie de bulle.
    system("pause");
    break;
case 6:
    tricomptage(tvaleur); //l'appel de la procedur qui fait le tri par comptage.
    cout<<endl<<endl;
    system("pause");
    break;
case 7:
    tripermutation(tvaleur); // l'appel de la procedur qui fait le tri de permutation.
    cout<<endl<<endl;
    system("pause");
    break;
case 9:
    system("cls");
    cout<<"Pour quitter le programme Appuyez sur une touche !!!"<<endl; /*
choix de quitter le programme.*/
    cout<<endl<<endl;
    getch();
    break;
default:;
case 8:
    int X;
    cout<<"Entrer un entier :";
    cin>>X;
    rechercher(tvaleur,X);

    cout<<endl<<endl;
    break;
}

}while(choice<9); // tant que la rep est ou on vas reboucler
}
```

Programmation structurée

```

/*****
***/

/***** Les fonction *****/

/*****
***/
int mpricipale() // fonction de menu pricipale //
{
    system("cls");
    int choix;
    cout<<"\t\t[ Le menu Principal ] "<<endl<<endl;
    cout<<" Choisisez une fonction que vous voulez appliquer"<<endl<<endl;
    cout<<"-1- remplir le tableau "<<endl;
    cout<<"-2- trouver le Min"<<endl;
    cout<<"-3- trouver le Max"<<endl;
    cout<<"-4- le trie par selection "<<endl;
    cout<<"-5- le trie de bulle"<<endl;
    cout<<"-6- le trie par comptage"<<endl;
    cout<<"-7- le trie de permutation"<<endl;
    cout<<"-8- chercher un element"<<endl;
    cout<<"-9- quitter le programme "<<endl;
    cin>>choix;
    return choix;
}
/*****
***/
/***** procedure d'affichage *****/
/*****
***/
void affichert(int affiche[],string message)
{

    cout<<message<<endl;
    for(int i=0;i<=9;i++)
        cout<<affiche[i]<<" /";

    cout<<endl<<endl<<endl;
}

/***** pour remplir le tableau *****/
/*****
***/
void remplir (int tremplir[]) // fonctoin de remplissage //
{
    system("cls");
    for(int i=0;i<=9;i++)
    {
        cout<<"Element "<<(i+1)<<" : ";
        cin>>tremplir[i];
    }
    system("cls");
    affichert(tremplir ," le tableau rempli est : ");
}
/*****
***/

/***** trouver le min *****/
/*****
***/
int emin (int temin[]) //fonction qui permet de trouver le Min //
```

```
{
    system("cls");
int min=temin[0];
    for (int i=0;i<=9;i++)
    {
        if (min>temin[i])
            min=temin[i];
    }

    affichert(temin, "le tableau d'origine est : ");
    return min;
}
//*****
***//

//*****le
max*****//
int emax ( int temax[])           //fonction qui permet de trouver le Max //
{
    system("cls");
int max=temax[0];
    for (int i=0;i<=9;i++)
    {
        if ( max<temax[i])
            max=temax[i];
    }
    affichert(temax, "le tableau d'origine est : ");
    return max;
}
//*****//

//*****trie par
selection*****//
void triselection ( int tselection[])           //fonction qui permet de faire le tri par selection//
{
    system("cls");

    affichert(tselection , "le tableau d'origine est : ");

    cout<<endl<<endl;
int tselec[10];
    memcpy(tselec,tselection,sizeof(tselec));
int pmin;
    for(int m=0;m<=8;m++)
    {
        int min=tselec[m];
        pmin=m;
        bool tri=true;
        for(int i=(m+1);i<=9;i++)
        {
            if(min>tselec[i])
            {
                min=tselec[i];
                pmin=i;
            }
        }
    }
}
```

```
        tri=false;
    }
}
if(tri==false)
{
    for(int j=(pmin);j>=(m+1);j--)
        tselec[j]=tselec[(j-1)];
    tselec[m]=min;
}
}

    affichert(tselec,"le tableau trieé par selection : ");

    cout<<endl;
}
//*****
****//

//***** trie de bulle
*****//
void tribulle(int tbulle[]) // fonction qui permet de faire le tri bulle //
{
    system("cls");
    affichert(tbulle , "le tableau d'origine est : ");
    cout<<endl;
    int tb[10];
    memcpy(tb,tbulle,sizeof(tb));
    int m=8;
    bool tt=true;
    do
    {
        tt=true;
        for(int i=0;i<=m;i++)
        {
            if(tb[i]>tb[(i+1)])
            {
                int num=tb[(i+1)];
                tb[(i+1)]=tb[i];
                tb[i]=num;
                tt=false;
            }
        }
        m=(m-1);
    }while ( tt==false||m<0 && m>0);

    affichert(tb , "le tableau trieé par bulle : ");
    cout<<endl;
}
//*****
****//

//***** trie par permutation
*****//
void tripermutation (int tpermutation[]) // fonction qui permet de faire le tri par
permutation //
```

```
{
    system("cls");
    affichert(tpermutation , "le tableau d'origine est : ");
    cout<<endl;
    int tperm[10];
    memcpy(tperm,tpermutation,sizeof(tperm));
    for (int f=0;f<=8;f++)
    {
        if ( tperm[f]>tperm[f+1]) //cette boucle permet de comparer les valeurs é trouvé le min
        {
            int i=0;
            int min=tperm[f+1];
            while (min>=tperm[i])
            {
                i=i+1;
            }
            for (int k=f+1;k>=i+1;k--)
            {
                tperm[k]=tperm[k-1];
            }
            tperm[i]=min;
        }
    }

    affichert(tperm , "le tableau trie par permutation : ");
    cout<<endl;
}
/*****a*****/

//***** trie par comptage
//*****//
void tricomptage ( int tcomptage[] )      //fonction qui permet de faire le trie par
comptage //
{
    system("cls");
    affichert(tcomptage , "le tableau d'origine est : ");

    int tcomp[10];
    memcpy(tcomp,tcomptage,sizeof(tcomp));
    int tposition[10]={0,0,0,0,0,0,0,0,0,0};
    for(int s=0;s<=9;s++)
    {
        for (int i=0;i<=9;i++)
        if (tcomp[s]>tcomp[i])
            tposition[s]=(tposition[s]+1); //donner l numéro de position de ce num c-a-d le nombre de
num plu min.
        else
        {
            if (tcomp[s]==tcomp[i] && s>i)
                tposition[s]=(tposition[s]+1);
        }
    }
}
bool tr;
do
{
```

```
for (int j=0;j<=8;j++)
{
    int num= tcomp[tposition[j]];
//*****//
    tcomp[tposition[j]]=tcomp[j];           // Partie de decalage sur le tableau des entier //
    tcomp[j]=num;                           // //
                                           // selon le tableau de position //
    int pos=tposition[tposition[j]];        //*****//
    tposition[tposition[j]]=tposition[j];
    tposition[j]=pos;
}
tr = false;
for (int k=0; k<=8; k++)
{
    if (tcomp[k]>tcomp[k+1])
        tr=true;
}
}while (tr==true);

affiche(tcomp, "le tableau trie par comptage est :");
cout<<endl;
}
//*****//
//***** rechercher un element
//*****//

//*****//
void rechercher(int trech[],int X)
{
    int pos=0;
    int i=0;
    do
    {
        if(trech[i]== X ) //tester si l'entier saisie est dans le tableau
        {
            pos=i+1;
        }

        i++;
    }while(i!=10 );

    system("cls");

    if(pos == 0)
        cout<<"l'element n'existe pas"<<endl; /*affichage pour dire que l'entier n'est existe
pas dans le tableau */
    else
        cout<<" l'element est existe , Sa position est : "<<pos<<endl;
    system("pause");
}
```

```
//***** Fin de Programme
//*****//
```

Exercice 55 :

```
#include <cstdlib>
#include <iostream>
#include <string>
#include <conio.h>
using namespace std;
```

```
// fonction des menu utiliser
int menu();
int menurech();
int menumodif();
```

```
void remplir();
```

```
// Les procedures du recherche
```

```
void rechercher();
void rechercher1();
```

```
void rechercher2();
```

```
void rechercher3();
```

```
// procedur
```

```
// du supression
```

```
void supprimer ();
```

```
// les Procedures de modification
```

```
void modifier1 ();
```

```
void modifier2();
```

```
void modifier3();
```

```
void modifier4();
```

```
void modifier5();
```

```
void modifier6();
```

```
void modifier7();
```

```
void modifier8();
```

```
void menuautre();
```

```
struct typelangue
{
string numinscri;
string nom;
string prenom;
string dateN;
string adresse;
string nivscolaire;
string Letudier;
int telephone;
}
```

```
int prix;           //
};

typelangue infolangue; // Declaration d1 variable avec mon propre type (typelangue)
typelangue tlangue[3]; //declaration d1 tableau avec mon propre type (typelangue)

/* declaration globale */
int bac =0;
int esp =0;
int engl=0;
int fr= 0;
int Prix = 0;

int main() // Voilà le main() pour executer notre travail
{
int choix;
do
{
    choix = menu();
    switch(choix)
    {
    case 1:
        remplir();
        break;
        getch();
    case 2:
        {
            do{

                int choix1 = menurech();
                switch (choix1)
                {
                    case 1:
                        rechercher1();
                        break;

                    case 2:
                        rechercher2();
                        break;

                    case 3:
                        rechercher3();
                        break;

                    case 4:
                        cout<<"<< Quitter le sous menu recherche ! >>"<<endl;
                        break;
                    default::
                }
                getch();
            }while(choix < 4);
        }
        break;

    case 3:
```

```
{
  int choix2 = menumodif();
  do{

      switch (choix2)
      {
          case 1:
              modifier1();
              break;

          case 2:
              modifier2();
              break;

          case 3:
              modifier3();
              break;
          case 4:
              modifier4();
              break;
          case 5:
              modifier5();
              break;
          case 6:
              modifier6();
              break;
          case 7:
              modifier7();
              break;
          case 8:
              modifier8();
              break;

          case 9:
              cout<<"<< Quitter le sous menu modifier ! >>"<<endl;
              system("pause");
              break;
          default:;
      }
      getch();
  }while(choix2<9);

  }
  break;
case 4:
  supprimer();
  break;
case 5:
  menuautre();
  break;

case 6:
  cout<<"<< Quitter le programme! >>"<<endl; //pour quitter le programme
  break;
  default;;
```



```
cin>>tlangue[i].Letudier;
string langue = tlangue[i].Letudier;
if("français" ==langue)

    fr = fr+1;

else
{
    if("anglais" ==langue)
    {
        engl = engl+1;
    }
    else
        esp =esp+1;
}

}

getch();
}

void menuautre()
{

    system("cls");
    cout<<"les bacheliers dans le centre :"<<bac<<" bacheliers "<<endl<<endl;
    cout<<"Les etudiants dans chaque langue :"<<endl;
    cout<<"*Français :"<<fr<<endl;
    cout<<"*Anglais  :"<<engl<<endl;
    cout<<"espagnole :"<<esp<<endl<<endl;
    cout<<"Le Prix total jusqu'a ce jour est : "<<Prix;
}

// rechercher1.....
int pos;
string nom,num ,lang;
void rechercher1()
{

    system("cls");
    bool sortir;
    sortir = false;
    int i=0;

    cout<<"Veuillez saisir le N~ d'inscription que vous voulez chercher "<<endl;
    cin>>num;

    for(int i=0; i<=2;i++)
    {
        if(num == tlangue[i].numinscri)
        {
            sortir = true;
            num = tlangue[i].numinscri;
            pos = i ;
        }
    }
}
```

```
}
if ( sortir == true)
{ system("cls");
cout<<"le N° d'inscription est :"<<tlangue[pos].numinscri<<endl;
cout<<"le nom : "<<tlangue[pos].nom<<endl;
cout<<"le prenom :"<<tlangue[pos].prenom<<endl;
cout<<" la date de naissance :"<<tlangue[pos].dateN<<endl;
cout<<" l'adresse :"<<tlangue[pos].adresse<<endl;
cout<<"le N° de tele :"<<tlangue[pos].telephone<<endl;
cout<<"le prix : "<<tlangue[pos].prix<<"Dh"<<endl;
cout<<"la niveau scolaire :"<<tlangue[pos].nivscolaire<<endl;
cout<<"la langue a etudier :"<<tlangue[pos].Letudier<<endl;
getch();
}

else
cout<<"Erreur";
getch();
}
////////// rechercher2//////////

void rechercher2()
{

system("cls");
bool sortir;
sortir = false;
int i=0;

cout<<"Veuillez saisir le nom que vous voulez chercher "<<endl;
cin>>nom;

for(int i=0; i<=2;i++)
{
if(nom == tlangue[i].numinscri)
{
sortir = true;
nom = tlangue[i].numinscri;
pos = i ;
}
}
if ( sortir == true)
{ system("cls");
cout<<"le N° d'inscription est :"<<tlangue[pos].numinscri<<endl;
cout<<"le nom : "<<tlangue[pos].nom<<endl;
cout<<"le prenom :"<<tlangue[pos].prenom<<endl;
cout<<" la date de naissance :"<<tlangue[pos].dateN<<endl;
cout<<" l'adresse :"<<tlangue[pos].adresse<<endl;
cout<<"le N° de tele :"<<tlangue[pos].telephone<<endl;
cout<<"le prix : "<<tlangue[pos].prix<<"Dh"<<endl;
cout<<"la niveau scolaire :"<<tlangue[pos].nivscolaire<<endl;
cout<<"la langue a etudier :"<<tlangue[pos].Letudier<<endl;
getch();
}
```

```
else
    cout<<"Erreur";
    getch();
}

////////// rechercher3
void rechercher3()
{
    system("cls");
    bool sortir;
    sortir = false;
    int i=0;

    cout<<"Veuillez saisir la langue que vous voulez chercher "<<endl;
    cin>>lang;

    for(int i=0; i<=2;i++)
    {
        if(nom == tlangue[i].numinscri)
        {
            sortir = true;
            nom = tlangue[i].numinscri;
            int pos = i ;
        }
    }
    if ( sortir == true)
    {system("cls");
    cout<<"le N° d'inscription est :"<<tlangue[pos].numinscri<<endl;
    cout<<"le nom : "<<tlangue[pos].nom<<endl;
    cout<<"le prenom :"<<tlangue[pos].prenom<<endl;
    cout<<" la date de naissance :"<<tlangue[pos].dateN<<endl;
    cout<<" l'adresse :"<<tlangue[pos].adresse<<endl;
    cout<<"le N° de tele :"<<tlangue[pos].telephone<<endl;
    cout<<"le prix : "<<tlangue[pos].prix<<"Dh"<<endl;
    cout<<"la niveau scolaire :"<<tlangue[pos].nivscolaire<<endl;
    cout<<"la langue a etudier :"<<tlangue[pos].Letudier<<endl;
    getch();
    }

    else
        cout<<"Erreur";
        getch();
    }

    //supprimer.....

    void supprimer()
    {
```

```
system("cls");
cout<<"N~ d'inscription "<<endl;
cin>>num;

bool sortir;
sortir = false;
for(int i=0; i<=2;i++)
{
    if(num == tlangue[i].numinscri)
    {
        sortir = true;
        tlangue[i].numinscri= "#";

        tlangue[i].nom.erase(0);

        tlangue[i].prenom.erase(0);

        tlangue[i].dateN.erase(0);

        tlangue[i].adresse.erase(0);

        tlangue[i].telephone =0;

        tlangue[i].prix = 0;

        tlangue[i].nivscolaire.erase(0);

        tlangue[i].Letudier.erase(0);
        getch();

        pos = i;
    }
}

if ( sortir == true)
{
    system("cls");
    cout<<"le N° d'inscription :"<<tlangue[pos].numinscri<<endl;
    cout<<"le nom :"<<tlangue[pos].nom<<endl;
    cout<<"le prenom :"<<tlangue[pos].prenom<<endl;
    cout<<"la date de naissance :"<<tlangue[pos].dateN<<endl;
    cout<<"l'adresse :"<<tlangue[pos].adresse<<endl;
    cout<<" Telephone :"<<tlangue[pos].telephone<<endl;
    cout<<" Prix :"<<tlangue[pos].prix<<endl;
    cout<<"Niveau Scolaire :"<<tlangue[pos].nivscolaire<<endl;
    cout<<"la langue :"<<tlangue[pos].Letudier<<endl;
    getch();

}
else
cout<<"Erreur";
getch();
}
```

```
///  
void modifier1()  
{  
  
    system("cls");  
    string name,pname,naissance,adresse,languetud,niveau;  
    int tele,prixx;  
  
    bool sortir;  
    sortir = false;  
    int i=0;  
  
    cout<<" N~ d'inscription "<<endl;  
    cin>>num;  
  
    for(int i=0; i<=2;i++)  
    {  
        if(num == tlangue[i].numinscri)  
        {  
            sortir = true;  
            num = tlangue[i].numinscri;  
            pos=i;  
        }  
    }  
    if ( sortir == true)  
    {  
        system("cls");  
        cout<<" nouveaux nom : "<<endl;  
        cin>>name;  
        tlangue[pos].nom = name;  
        cout<<"le N° d'inscription : "<<tlangue[pos].numinscri<<endl;  
        cout<<"le nom : "<<tlangue[pos].nom<<endl;  
        cout<<"le prenom : "<<tlangue[pos].prenom<<endl;  
        cout<<"la date de naissance : "<<tlangue[pos].dateN<<endl;  
        cout<<"l'adresse : "<<tlangue[pos].adresse<<endl;  
        cout<<" Telephone : "<<tlangue[pos].telephone<<endl;  
        cout<<" Prix : "<<tlangue[pos].prix<<endl;  
        cout<<"Niveau Scolaire : "<<tlangue[pos].nivscolaire<<endl;  
        cout<<"la langue : "<<tlangue[pos].Letudier<<endl;  
        getch();  
    }  
    else  
    cout<<"Erreur";  
    getch();  
}  
  
/////modifier2  
void modifier2()  
{  
  
    system("cls");  
    string pname;
```

```
bool sortir;
sortir = false;
int i=0;

cout<<"N~ d'inscription "<<endl;
cin>>num;

for(int i=0; i<=2;i++)
{
    if(num == tlangue[i].numinscri)
    {
        sortir = true;
        num = tlangue[i].numinscri;
        pos =i;
    }
}
if ( sortir == true)
{system("cls");
cout<<" nouveaux prenom :"<<endl;
cin>>pname;
tlangue[pos].prenom = pname;
cout<<"le N° d'inscription :"<<tlangue[pos].numinscri<<endl;
cout<<"le nom :"<<tlangue[pos].nom<<endl;
cout<<"le prenom :"<<tlangue[pos].prenom<<endl;
cout<<"la date de naissance :"<<tlangue[pos].dateN<<endl;
cout<<"l'adresse :"<<tlangue[pos].adresse<<endl;
cout<<" Telephone :"<<tlangue[pos].telephone<<endl;
cout<<" Prix :"<<tlangue[pos].prix<<endl;
cout<<"Niveau Scolaire :"<<tlangue[pos].nivscolaire<<endl;
cout<<"la langue :"<<tlangue[pos].Letudier<<endl;
getch();
}
else
cout<<"Erreur";
getch();
}

////modifier 3
void modifier3()
{
    system("cls");
    string naissance;
    bool sortir;
    sortir = false;
    int i=0;

    cout<<"N~ d'inscription "<<endl;
    cin>>num;

    for(int i=0; i<=2;i++)
    {
```

```
    if(num == tlangue[i].numinscri)
    {
        sortir = true;
        num = tlangue[i].numinscri;
        pos =i;
    }
}
if ( sortir == true)
{system("cls");
cout<<" nouveau date de naissance :"<<endl;
cin>>naissance;
tlangue[pos].dateN= naissance;
cout<<"le N° d'inscription :"<<tlangue[pos].numinscri<<endl;
cout<<"le nom :"<<tlangue[pos].nom<<endl;
cout<<"le prenom :"<<tlangue[pos].prenom<<endl;
cout<<"la date de naissance :"<<tlangue[pos].dateN<<endl;
cout<<"l'adresse :"<<tlangue[pos].adresse<<endl;
cout<<" Telephone :"<<tlangue[pos].telephone<<endl;
cout<<" Prix :"<<tlangue[pos].prix<<endl;
cout<<"Niveau Scolaire :"<<tlangue[pos].nivscolaire<<endl;
cout<<"la langue :"<<tlangue[pos].Letudier<<endl;
getch();
}
else
cout<<"Erreur";
getch();
}

/////modifier 4
void modifier4()
{
    system("cls");
    string adresse;
    bool sortir;
    sortir = false;
    int i=0;

    cout<<" N~ d'inscription " <<endl;
    cin>>num;

for(int i=0; i<=2;i++)
{
    if(num == tlangue[i].numinscri)
    {
        sortir = true;
        num = tlangue[i].numinscri;
        pos=i;
    }
}
if ( sortir == true)
{
    system("cls");
    cout<<" nouveau Adresse :"<<endl;
    cin>>adresse;
    tlangue[pos].adresse = adresse;
```

```
cout<<"le N° d'inscription :"<<tlangue[pos].numinscri<<endl;
cout<<"le nom :"<<tlangue[pos].nom<<endl;
cout<<"le prenom :"<<tlangue[pos].prenom<<endl;
cout<<"la date de naissance :"<<tlangue[pos].dateN<<endl;
cout<<"l'adresse :"<<tlangue[pos].adresse<<endl;
cout<<" Telephone :"<<tlangue[pos].telephone<<endl;
cout<<" Prix :"<<tlangue[pos].prix<<endl;
cout<<"Niveau Scolaire :"<<tlangue[pos].nivscolaire<<endl;
cout<<"la langue :"<<tlangue[pos].Letudier<<endl;
getch();
}
else
cout<<"Erreur";
getch();

//modifier 5
}
void modifier5()
{

    system("cls");
    int tele;
    bool sortir;
    sortir = false;
    int i=0;

    cout<<" N~ d'inscription "<<endl;
    cin>>num;

for(int i=0; i<=2;i++)
{
    if(num == tlangue[i].numinscri)
    {
        sortir = true;
        num = tlangue[i].numinscri;
        pos = i;
    }
}
if ( sortir == true)
{system("cls");
cout<<" nouveaux telephone :"<<endl;
cin>>tele;
tlangue[pos].telephone = tele;
cout<<"le N° d'inscription :"<<tlangue[pos].numinscri<<endl;
cout<<"le nom :"<<tlangue[pos].nom<<endl;
cout<<"le prenom :"<<tlangue[pos].prenom<<endl;
cout<<"la date de naissance :"<<tlangue[pos].dateN<<endl;
cout<<"l'adresse :"<<tlangue[pos].adresse<<endl;
cout<<" Telephone :"<<tlangue[pos].telephone<<endl;
cout<<" Prix :"<<tlangue[pos].prix<<endl;
cout<<"Niveau Scolaire :"<<tlangue[pos].nivscolaire<<endl;
cout<<"la langue :"<<tlangue[pos].Letudier<<endl;
getch();
}
else
```

```
cout<<"Erreur";
getch();

}
/////modifier 6
void modifier6()
{

    system("cls");
    int prixx;
    bool sortir;
    sortir = false;
    int i=0;

    cout<<" N~ d'inscription :"<<endl;
    cin>>num;

for(int i=0; i<=2;i++)
{
    if(num == tlangue[i].numinscri)
    {
        sortir = true;
        num = tlangue[i].numinscri;
        pos=i;
    }
}
if ( sortir == true)
{
    system("cls");
    cout<<" nouveaux Prix :"<<endl;
    cin>>prixx;
    tlangue[pos].prix = prixx;
    cout<<"le N° d'inscription :"<<tlangue[pos].numinscri<<endl;
    cout<<"le nom :"<<tlangue[pos].nom<<endl;
    cout<<"le prenom :"<<tlangue[pos].prenom<<endl;
    cout<<"la date de naissance :"<<tlangue[pos].dateN<<endl;
    cout<<"l'adresse :"<<tlangue[pos].adresse<<endl;
    cout<<" Telephone :"<<tlangue[pos].telephone<<endl;
    cout<<" Prix :"<<tlangue[pos].prix<<endl;
    cout<<"Niveau Scolaire :"<<tlangue[pos].nivscolaire<<endl;
    cout<<"la langue :"<<tlangue[pos].Letudier<<endl;
    getch();
}
else
cout<<"Erreur";
getch();

}
/////modifier 7

void modifier7()
{

    system("cls");
    string niveau;
    bool sortir;
```

```
    sortir = false;
    int i=0;

    cout<<" N° d'inscription " <<endl;
    cin>>num;

for(int i=0; i<=2;i++)
{
    if(num == tlangue[i].numinscri)
    {
        sortir = true;
        num = tlangue[i].numinscri;
        pos =i;
    }
}
if ( sortir == true)
{system("cls");
cout<<" nouveau Niveau Scolaire : " <<endl;
cin>>niveau;
tlangue[pos].nivscolaire = niveau;
cout<<"le N° d'inscription : " <<tlangue[pos].numinscri<<endl;
cout<<"le nom : " <<tlangue[pos].nom<<endl;
cout<<"le prenom : " <<tlangue[pos].prenom<<endl;
cout<<"la date de naissance : " <<tlangue[pos].dateN<<endl;
cout<<"l'adresse : " <<tlangue[pos].adresse<<endl;
cout<<" Telephone : " <<tlangue[pos].telephone<<endl;
cout<<" Prix : " <<tlangue[pos].prix<<endl;
cout<<"Niveau Scolaire : " <<tlangue[pos].nivscolaire<<endl;
cout<<"la langue : " <<tlangue[pos].Letudier<<endl;
getch();
}
else
cout<<"Erreur";
getch();
}
////modifier 8
void modifier8()
{

    system("cls");
    string languetud;
    bool sortir;
    sortir = false;
    int i=0;

    cout<<"Veuillez saisir le N° d'inscription que vous voulez modifier?" <<endl;
    cin>>num;

for(int i=0; i<=2;i++)
{
    if(num == tlangue[i].numinscri)
    {
        sortir = true;
        num = tlangue[i].numinscri;
```



```
    }while(tab1[i-1]!=tab2[j-1] && tab2[j-1]>5);  
  
    system("PAUSE");  
  
}
```

Exercice 57 :

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction  
C system()*/  
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie  
*****/  
using namespace std; //espace de nom  
int main(int argc, char *argv[]) //le MAINE  
{  
    //declaration des tableau de type char  
    char tab[]={'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','a'};  
    char phrase[50];  
  
    //saisie de la phrase qu'on veut la coder  
  
    cout<<"entrer une chaine : ";  
    cin.getline(phrase,50);  
    int X = strlen(phrase);  
    system("cls");  
  
    cout<<"la chaine : "<<phrase<<" , sera : ";  
  
    int j, i=-1;  
    // la boucle du cryptage  
    do  
    {  
        i++;  
        for(j=0;j<=26;j++)  
        {  
            if(phrase[i] == tab[j] && phrase[i]!=' ' )  
            {  
                phrase[i]=tab[j+1];  
                cout<<phrase[i];  
                break;  
            }  
        }  
    }  
  
    if(phrase[i]==' ' )  
        cout<<" ";  
  
    }while(i!=X );  
  
    cout<<endl<<endl;  
    system("PAUSE");  
  
}
```

Exercice 58 :

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /***** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
    //déclaration des tableaux de type char
    Char tab[]={ 'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t',
'u','v','w','x','y','z','a','b','c','d','e','f','g','h','i','j','k','l','m','n'};
    char phrase[50];
    int dec; //déclaration d'un entier de type char

    //saisie de la phrase qu'on veut la coder
    cout<<"entrer une chaine : ";
    cin.getline(phrase,50);
    int X = strlen(phrase);

    do
    {
        cout<<"Entrer un entier pour le decalage (entre 1 et 13)"<<endl;
        cin>>dec;
    }while(dec<1 ||dec>13);

    system("cls");
    cout<<"la chaine : "<<phrase<<" , sera : ";

    int j, i=-1;
    // la boucle du cryptage
    do
    {
        i++;
        for(j=0;j<=25;j++)
        {
            if(phrase[i] == tab[j] && phrase[i]!=' ')
            {

                phrase[i]=tab[dec+j];
                cout<<phrase[i];
                break;

            }

        }

        if(phrase[i]==' ')
            cout<<" ";

    }while(i!=X );
    cout<<endl<<endl;
    system("PAUSE");
}
```

Exercice 59 :

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
    //déclaration des tableaux de type char
    char
    tab[]={ 'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','a','b','c','d','e','f','g',
    'h','i','j','k','l','m','n'};
    char phrase[50];
    char
    code[]={ 'H','Y','L','U','J','P','V','R','E','A','K','B','N','D','O','F','S','Q','Z','C','W','M','G','T','T','X'};

    //saisie de la phrase qu'on veut la coder
    cout<<"entrer une chaine : ";
    cin.getline(phrase,50);
    int X = strlen(phrase);
    system("cls");

    cout<<"la chaine : "<<phrase<<" , sera : ";

    // la boucle du cryptage
    int j, i=-1;
    do
    {
        i++;
        for(j=0;j<=26;j++)
        {
            if(phrase[i] == tab[j] && phrase[i]!=' ')
            {
                phrase[i]=code[j];
                cout<<phrase[i];
                break;
            }
        }

        if(phrase[i]==' ')
            cout<<"/";

    }while(i!=X );

    cout<<endl<<endl;
    system("PAUSE");
}
```

Exercice 60 :

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
```

```
{

char phrase[50]; //déclaration d'un tableau de type char
char C; //déclaration d'un variable de type char

//saisie de la chaîne
cout<<"Entrer une chaine : ";
cin.getline(phrase,50);

int X = strlen(phrase);
cout<<"Entrer le caracter : ";
cin>>C;

int i=0,nbr=0;
//On cherche l'occurrence qui existe et on le supprime
do
{

for(int i=0; i<X;i++)
{
if(C==phrase[i])
{

for(int j=i; j<X;j++)
{
phrase[i]=phrase[i+1];

}

}

}

i++;
}while(i<X);

for(int i=0 ;i<=X;i++)
cout<<phrase[i];

cout<<endl<<endl;
system("PAUSE");

}
```

Exercice 61 :

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
void doublon(char tab[],char t[],int &l) // une procédure qui cherche les doublons dans une
chaîne
{ int x,j,i,a,l=0;
t[0]=tab[0];
for(i=0;i<20;i++)
{ a=0;j=0;
```

```
do
{
  if(tab[i]==t[j])
  {
    a++;
  }
  j++;
}while(j<l && tab[i]!=t[j]);
if(a==0)
{
  t[l+1]=tab[i];
  l++;
}
}
}
int main(int argc, char *argv[]) //le MAINE
{
  char tab[20],t[20];//déclaration des tableaux de type char
  int i,l;//déclaration des variables de type entier

  //saisie de la chaîne
  cout<<"saisir une phrase :";
  cin.getline(tab,20);
  doublon(tab,t,l); //l'appel de la procédure
  cout<<"la phrase avec supprimant les doublons :";
  for(i=0;i<l;i++)
  {
    cout<<t[i];
  }
  system("PAUSE");
  return EXIT_SUCCESS;
}
```

Exercice 62 :

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{

  char t1[50],t2[50];//déclaration des tableaux de type char
  int j,i=-1,lg=0;//déclaration des variables de type entier

  //saisie de la 1er chaîne
  cout<<"veuillez saisir un phrase "<<endl;
  cin.getline(t1,49);
  int K = strlen(t1);
//saisie de la 2eme chaîne
  cout<<"veuillez saisir une deuxieme phrase "<<endl;
  cin.getline(t2,49);
  int H = strlen(t2);
  //Une boucle pour chercher si les deux chaînes sont equivalentes
  do
```

```
{
    i++;
    for( j=0;j<=H;j++)
    {
        if (t1[i]==t2[j])
        {
            lg=1;
            break;
        }
    }
}while(t1[i]!=t2[j] && i!=50);

if(lg==1) //si elles sont equivalentes
cout<<"les deux phrase sont equivalent "<<endl;
else //si non
cout<<"les deux phrase ne son pas equivalent"<<endl;

system("PAUSE");
return EXIT_SUCCESS;
}
```

Exercice 63:

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
    char tab[50],mot[20]; //déclaration des tableaux de type char
    bool trouve;//déclaration des variables de type booleéne

    //saisie de la chaîne

    cout<<"Taper une phrase terminer avec '.'"<<endl;
    cin.getline(tab,50);
    int X = strlen(tab);
    //Saisie du mot
    cout<<"Taper un mot "<<endl;
    cin.getline(mot,20);
    int Y = strlen(mot);

    int nbr=0;
    int j=0;
    int i=0;
    do
    {
        // Une boucle pour vérifier si les lettres du mot sont bien présentes dans la phrase
        do
        {
            trouve=false;
            if(tab[i]== mot[j] && tab[i]!=' ')
            {

                nbr++;
            }
        }
    }
}
```

```
    trouve=true ;
  }
  i++;
  }while(i-1==X && trouve == true );
  j++;
}while(j!=Y );

if(nbr==X) //si les lettres du mot sont bien présentes dans la phrase
cout<<"Le Mot : "<<mot<<" est eparpillement"<<endl;
else //si non
cout<<"Le Mot : "<<mot<<" est Non-eparpillement"<<endl;

system("PAUSE");

}
```

Exercice 64 :

```
/*
Exemple :  esope      reste      ici      et      se      repose.
*/

#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
    char phrase[80]; //déclaration d'un tableau de type char

    //saisie de la chaîne
    cout<<"Entrer une chaine: ";
    cin.getline(phrase,80);
    int l=strlen(phrase);
    bool pln=true; //Declaration d'une variable de type booleenne et l'initialiser par vrai
    //Une boucle pour voir si cette phrase peut se lire dans les deux sens.

    for(int i=0;i<=l/2;i++)
    {
        if(pln==true)
        {
            if(phrase[i]!=phrase[l-i-1])
            {
                pln=false;
            }
        }
        else
        {
            break;
        }
    }
    if(pln==true) //si cette phrase peut se lire dans les deux sens.

    {
        cout<<"Palindrome";
    }
}
```

```
    }
    Else //si non
    {
        cout<<"Non Palindrome";
    }
    system("pause");
}
```

Exercice 65 :

```
/*
On veut calculer  $K=1 * 1/2 * 1/3 * \dots * 1/n(n>0)$ .
*/

#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/

using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
    int N; //déclaration de variable de type entier

    cout<<"Veuillez saisir une entier : " ; //affichage de saisir
    cin>>N; //la lecture

    float k=1; //déclaration de variable de type réel et l'initialiser par 1

    cout<<"K = ";
    //Une boucle pour fair le calcul de K
    for(float i=1;i<N;i++)
    {
        cout<<1<<" "<<i;
        if(i<N-1)
            cout<<"*";
        k=(1/i);
        k=k*k;
    }
    cout<<endl<<"K = "<<k<<endl;
    system("PAUSE");
}
```

Exercice 66 :

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/

using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
    int Mnt ; //déclaration de variable de type entier
    cout<<"veuillez saisir le montant d'achat ";
    cin>>Mnt;
    if(Mnt>5000) // si le montant d'achat de plus de 5000 dhs
    {
```

```
Mnt= Mnt-(Mnt*20/100);
cout<<"votre reduction est de 20%, donc : "<<Mnt<<" Dhs";
}
Else // si non
{
if(Mnt>3000 ) // si le montant d'achat est > 3000 dhs
{
Mnt=Mnt-(Mnt*(15/100));
cout<<"votre reduction est de 15%, donc : "<<Mnt<<" Dhs";
}
Else //si non
{
if(Mnt>1000 ) //si le montant d'achat est > 1000 dhs
{
Mnt=Mnt-(Mnt*10/100);
cout<<"votre reduction est de 10%, donc : "<<Mnt<<" Dhs";
}
Else //si non
cout<<"Aucune reduction pour votre montant";
}
}
}
system("PAUSE");
}
```

Exercice 67 :

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
int prix,cmd,Mnt; int Mnt ; //déclaration des variables de type entier
cout<<" Entrer le prix Unitaire : ";cin>>prix;
cout<<" Entrer la quantité de commande : ";cin>>cmd;
Mnt = prix * cmd;
system("cls");
if(Mnt>1000) //si le montant hors taxe est supérieur à 1000 dh
{
Mnt=Mnt-(Mnt*10/100);
cout<<"Le port est gratuit "<<endl;
cout<<"la remise est de 10% "<<endl;
cout<<"Le prix a payer : "<<Mnt<<" Dhs";
}
Else // si non
{
if(Mnt<1000) //si le montant hors taxe est inferieur à 1000 dh
{
cout<<"Le port est de 3% "<<endl;
if(Mnt>300) si le montant hors taxe est compris entre 300 et 1000
{
Mnt=Mnt-(Mnt*5/100);
cout<<" la remise est de 5%"<<endl;
}
}
cout<<"Le prix a payer : "<<Mnt<<" Dhs";
}
```

```

    }
  }
  cout<<endl<<endl;
  system("PAUSE");
}

```

Exercice 69 :

```

/*
On veut faire le tri dans l'ordre croissant et décroissant d'une matrice de taille N × M
*/

```

```

#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/

```

```

using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{

```

```

  int j,i,k,min,m; // declaration
  bool sortir; // declaration
  int tb[2][3]; // declaration

```

```

  for(i=0;i<2;i++) //l'utilisation de la boucle for permet de remplir le tableau tb[]
  {
  for(j=0;j<3;j++) //l'utilisation de la boucle for permet de remplir le tableau tb[]
  {
  cout<<"Element "<<i<<","<<j<<" : ";
  cin>>tb[i][j];
  }
  }

```

```

  j=2; //initialisation du variable j par une valeur fixe
  do
  {
  //
  //
  sortir = true; // T R I
  for(i=0;i<=j;i++) //
  {
  // T R I
  for(k=0;k<= j;k++) //
  {
  //
  // P A R
  // B U L L E //
  if(tb[i][k]>tb[i][k+1]) //
  {
  min = tb[i][k+1]; //
  tb[i][k+1] = tb[i][k]; //
  tb[i][k]= min; //
  sortir = true; //
  }
  }
  }
  j--;
  }while(j==0 && sortir == true);
  system("cls");
  cout<<"la matrice trier dans l'ordre croissant : "<<endl;
  // Afficher le resultat du tri
  for(i=0;i<2;i++)
  {
  for(j=0;j<3;j++)

```

```
{
  cout<<tb[i][j]<<" ";
}
cout<<endl;
}
/*****/
cout<<endl<<"la matrice trier dans l'ordre decroissant : "<<endl;
// Afficher le resultat du tri
for(i=1;i>=0;i--)
{
  for(j=2;j>=0;j--)
  {
    cout<<tb[i][j]<<" ";
  }
  cout<<endl;
}
system("pause");
}
```

Exercice 70 :

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /***** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
  int i,Suite[7]; // declaration
  Suite[0]=1; // initialisation
  Suite[1]=1; // initialisation
  for(i=2;i<7;i++){
    Suite[i]=Suite[i-1]+Suite[i-2];
  }
  cout<<"  Voici votre suite :"<<endl<<endl;
  for(i=0;i<6;i++){
    cout<<Suite[i]<<" / ";
  }
  cout<<Suite[6]<<" . ";
  cout<<endl<<endl<<endl;
  system("pause");
  return0;
}
```



```
Voici votre suite :
1 / 1 / 2 / 3 / 5 / 8 / 13 .
Appuyez sur une touche pour continuer...
```

Exercice 71 :

```
/*
- Saisir les données nécessaires
- Calculer la moyenne pour chaque stagiaire
- Trier les stagiaires par la moyenne et dans le sens décroissant.
- Déterminer le classement pour chaque stagiaire.
Afficher les données de tous les stagiaires
*/

#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/

using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
int i,j,k,pos,coef,coef1;
int tab[5];
string Tnom[5]; // declaration d un tableau de type chaine de caractere
string Tprenom[5]; // declaration d un tableau de type chaine de caractere
float S, Tmoy[5],min,Tnote[5]; //declaration d'un tableau de type réel
float note =1;
float note1 =0;
for(i=0; i<=4; i++)
{
cout<<"//-----//"<<endl;
cout<<"\t Stagiaire N:"<<i+1;
cout<<endl<<endl; //-----//
// Affichage //
cout<<"Nom :"; // de saisit //
cin>>Tnom[i];
cout<<"prenom :";
cin>>Tprenom[i];
coef1 =0;
note1=0;
for(j=0; j<=4; j++)
{
// pour remplir //
cout<<"Note : "<<i+1; // le tableau //
cin>>Tnote[i];
cout<<"Coef : "<<i+1;
cin>>coef;
//-----//
note=coef*Tnote[i];
note1=note+note1;
coef1=coef+coef1;
}
Tmoy[i]= note1/coef1; //calculer la somme des entiers
}
for(i=0;i<5;i++)
{
pos=i;
min=Tmoy[i];
for(j=i+1;j<5;j++)
{
if(Tmoy[j]>min)
{
min=Tmoy[j];
}
}
}
}
}
```

```

        pos=j;
    }
}
for(k=pos;k>=i+1;k--)
{
    Tmoy[k]=Tmoy[k-1];
}
Tmoy[i]=min;
}
/* affichage du tableau */
system("cls");
for(i=0;i<5;i++)
{
cout<<"Classement : "<<i+1<<endl;
    cout<<"Nom :";           //      de saisit           //
    cout<<Tnom[i]<<endl;
    cout<<"prenom :";
    cout<<Tprenom[i]<<endl;

    for(j=0; j<=4; j++)
    {
        cout<<"Note : "<<i+1;           //           pour remplir           //
        cout<<Tnote[i]<<endl;           //           le tableau           //
    }
    cout<<"Moyenne : "<<Tmoy[i]<<endl<<endl;
}
system("PAUSE");
}

```

Exercice 72 :

```

/*
Si s = [15,4,19,4,8,11,11,3,4,19] et L = 10 alors s'=[15,4,19,8,11,3] et L=6
*/
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
    int i,j,L; // declaration
    int T[10]; // declaration
    int S[10]; // declaration
    for(i=0;i<9;i++)
    {
        cout<<"Element("<<i+1<<"): ";
        cin>>T[i];
    }
    for(i=0;i<9;i++)
        S[i]=T[i];
    int nbr=0;
    do
    {
        for(i=0;i<10;i++)
        {
            for(j=0;j<10;j++)

```

```
{
  if(T[i] == S[j])
  {
    nbr++;
    for(int k=i;k<10;k++)
      T[i]=T[i+1];
  }
}
}
// Affichage du resultat
}while(j<10 && T[i]!=S[j]);
cout<<"s'=[";
for(i=0;i<10;i++)
cout<<T[i]<<" ";
cout<<"]"<<endl;
cout<<"l'="<<10-nbr<<endl;
system("PAUSE");
}
```

Exercice 73 :

```
/*
L'utilisateur doit taper 10 entiers qui seront stockés dans un tableau.
Ensuite on affiche soit "le tableau est croissant", soit "le tableau est décroissant", soit "le
tableau est constant", soit "le tableau est quelconque".
*/
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
  int a[N],i;
  bool croissant=true,
  bool decroissant=true;

  for(i=0;i<N;i++)
  {
    cout<<"Veuillez taper l'entier numero "<<i<<" : ";
    cin>>a[i];
  }
  for(i=0;i<N-1;i++)
  {
    if(a[i]>a[i+1])croissant=false;
    if(a[i]<a[i+1])decroissant=false;
  }
  if(croissant==true && decroissant==true) // si le tableau est constant
    cout<<"le tableau est constant"<<endl;
  else{
    if(croissant==true && decroissant==false) // si le tableau est croissant
      cout<<"le tableau est croissant"<<endl;
    else{
      if(croissant==false && decroissant==true) // si le tableau est decroissant
        cout<<"le tableau est decroissant"<<endl;
      else // si le tableau est quelconque
        cout<<"le tableau est quelconque"<<endl;
    }
  }
}
```

```
    }
    system("pause");
    return 0;
}
```

Exercice 74:

```
/* On veut transformer toutes les minuscules de la chaîne en majuscules.*/
#include <cstdlib> /*On inclure la bibliothèque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliothèque des entrées/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
    char tab[50]; // Déclaration
    int k; // Déclaration
    cout << "Tapez une chaîne svp : ";
    k=-1;
    do{
        k++;
        tab[k]=getchar();
    }while(tab[k]!='.' && k<49);
    int i=0;
    while(tab[i]!='.' && i<49)
    {
        if(tab[i]>='a' && tab[i]<='z'){
            tab[i]=tab[i]+'A'-'a';
        }
        i++;
    }
    cout<<"La chaîne finale est : "<<endl;
    k=-1;
    do{
        k++;
        cout<<tab[k];
    }while(tab[k]!='.' && k<49);
    cout<<endl;
    system("pause");
    return 0;
}
```

Exercice 75:

```
/* On veut supprimer toutes les voyelles saisies par l'utilisateur.*/
#include <cstdlib> /*On inclure la bibliothèque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliothèque des entrées/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
    char a[50]; //Déclaration d'un tableau de type caractère
    int b,k,nb; // Déclaration des variables de type entier
    cout << "Tapez une chaîne svp : ";
    k=-1;
    do{ //Une boucle pour saisir la phrase
        k++;
```

```
a[k]=getchar();
}while(a[k]!='.' && k<49);

nb = strlen(a); //Une instruction pour connaitre la taille du tableau
int i=0,j;
while(a[i]!='.' && i<49)
{
    if(a[i]=='a' || a[i]=='A') { //Si la voyelle egale à “a” ou bien “A”
        nb--;
        for (j=i; j<nb; j++){
            a[j]=a[j+1];
            a[nb]='\0';
            i--;}
        }

    if(a[i]=='e' || a[i]=='E') { //Si la voyelle egale à “e” ou bien “E”
        nb--;
        for (j=i; j<nb; j++) {
            a[j]=a[j+1];
            a[nb]='\0';
            i--;}
        }

    if(a[i]=='i' || a[i]=='I') { //Si la voyelle egale à “i” ou bien “I”
        nb--;
        for (j=i; j<nb; j++) {
            a[j]=a[j+1];
            a[nb]='\0';
            i--;}
        }

    if(a[i]=='o' || a[i]=='O') { //Si la voyelle egale à “o” ou bien “O”
        nb--;
        for (j=i; j<nb; j++){
            a[j]=a[j+1];
            a[nb]='\0';
            i--;}
        }

    if(a[i]=='u' || a[i]=='U') { //Si la voyelle egale à “u” ou bien “U”
        nb--;
        for (j=i; j<nb; j++) {
            a[j]=a[j+1];
            a[nb]='\0';
            i--;}
        }

    if(a[i]=='y' || a[i]=='Y') { //Si la voyelle egale à “y” ou bien “Y”
        nb--;
        for (j=i; j<nb; j++) {
            a[j]=a[j+1];
            a[nb]='\0';
            i--;}
        }

    i++;
}
cout << "La chaine finale est :" << endl; //affichage de la chaine après la suppression des
voyelles
k=-1;
```

```
do{
    k++;
    cout <<a[k]<< endl;
    }while(a[k]!='.' && k<49);

    system("pause");
    return 0;
}
```

Exercice 76:

```
/* On veut conjugais un verb de 1er groupe saisie par l'utilisateur à l'indicatif présent.*/
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
int main(int argc, char *argv[]) //le MAINE
{
    char tab[20]; //Declaration d'un tableau de type caractere
    cout << "Tapez un verbe du premier groupe : ";
    cin >> tab;
    int i;
    i = strlen(tab);
    if (i<=2 || tab[i-1]!='r' || tab[i-2]!='e') //une condition pour tester si le verb se termine par er
à la fin
        cout << "le verbe n'est pas du premier groupe" << endl;
    else // si il se termine par "er"
    {
        tab[i-2]='\0';
        cout<<"-----"<<endl;
        cout << "je | " << tab << "e" << endl;
        cout<<"-----"<<endl;
        cout << "tu | " << tab << "es" << endl;
        cout<<"-----"<<endl;
        cout << "il | " << tab << "e" << endl;
        cout<<"-----"<<endl;
        cout << "nous | " << tab << "ons" << endl;
        cout<<"-----"<<endl;
        cout << "vous | " << tab << "ez" << endl;
        cout<<"-----"<<endl;
        cout << "ils | " << tab << "ent" << endl;
        cout<<"-----"<<endl;
    }
    system("pause");
    return 0;
}
```

Exercice 77:

```
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie
*****/
using namespace std; //espace de nom
//Declaration des fonctions et des procedures
void Liste(string liste[],int X);
int Ajou(string liste[],string code[],int nbr,int Qnt[]);
```

```
void suppr(string liste[],string code[],int X);
void Ach(string liste[],string Code[],int prix[],int X);
void Vnd(string liste[],string Code[],int prix[],int X);

int main(int argc, char *argv[])
{
    string rep , liste[10];
    string Code[10];
    int Qnt[10];
    int prix[10]={ 100,101,105,50,180,200,320,150,172,111};
    int choix,X,nbr;
do
{
    //Affichage du menu
    cout<<"/-/-/-/-/_-[MENU]-_/_/_/_/_/"<<endl<<endl;
    cout<<"t Entrer votre choix!"<<endl<<endl;
    cout<<"-1- Ajouter un produit "<<endl;
    cout<<"-2- Afficher la liste de produits"<<endl;
    cout<<"-3- Supprimer un produit"<<endl;
    cout<<"-4- Acheter un produit"<<endl;
    cout<<"-5- Vendre un produit"<<endl;
    cout<<"-6- Quitter"<<endl<<endl;
    cout<<"SVP saisie votre choix : ";
    cin>>choix;
    switch(choix)
    {
        case 1:
            nbr=0;
            X = Ajou(liste,Code,nbr,Qnt);
            break;
        case 2:

            Liste(liste , X);
            break;
        case 3:
            suppr(liste,Code,X);
            break;
        case 4:
            Ach(liste,Code,prix,X);
            break;
        case 5:
            Vnd(liste,Code,prix,X);
            break;
        case 6:
            default::;
            break;

    }
    cout<<"Recommencer ?";
    cin>>rep;
}while(rep=="oui");
getch();
}

//////////////////////////////// la liste des produit //////////////////////////////////
```

```
void Liste(string liste[],int X)
{

system("cls");
for(int i=0;i<=X-1;i++)
cout<<"(" <<i+1<<" ) "<<liste[i]<<endl;

}
////////// Ajouter un nouveau produit //////////
int Ajou(string liste[],string code[],int nbr ,int Qnt[])
{
system("cls");
int i=0;
string rep;
do
{
cout<<"Entrer le Code de produit : ";cin>>code[i];
cout<<"Entrer l'intitulé de produit : ";cin>>liste[i];
do
{
cout<<"Entrer la quantite : ";cin>>Qnt[i];
}while(Qnt[i]==0);
i++;
nbr++;
system("cls");
cout<<endl<<endl<<"Ajouter un autre produit ? oui/non"<<endl;
cin>>rep ;
}while(rep=="oui");

return nbr;
}

////////// supprimer un produit appartir de son code//////////
void suppr(string liste[] ,string code[], int X)
{
system("cls");
string cod;
cout<<"Code : ";
cin>>cod;
int i=-1;
do
{
i++;
if(cod==code[i])
{
code[i]="0";
liste[i]="0";

}

}while(cod==code[i] && i==X);

}

}
```

```
//////////Achetre un produit //////////
void Ach(string liste[],string Code[],int prix[],int X)
{
    system("cls");
    string cod;
    int qnt,i;
    i=-1;
    cout<<"Code Produit : ";
    cin>>cod;

    do
    {
        i++;
        if(cod == Code[i])
        {
            cout<<"Quantitee achetee : ";cin>>qnt;

            system("cls");

            cout<<endl<<endl<<"Code :"<<Code[i]<<"\t\t Produit : "<<liste[i]<<"\t\t Quantiter :
"<<qnt<<endl;
            qnt = qnt * prix[i];
            cout<<"Le prix : "<<qnt;

        }
    }while(cod!=Code[i] && i<=X);
}

////////// Vendre un produit
void Vnd(string liste[],string Code[],int prix[],int X)
{
    system("cls");
    string cod;
    int qnt,i;
    i=-1;
    cout<<"Code Produit : ";
    cin>>cod;

    do
    {
        i++;
        if(cod == Code[i])
        {
            cout<<"Quantitee vendu : ";cin>>qnt;

            system("cls");

            cout<<endl<<endl<<"Code :"<<Code[i]<<"\t\t Produit : "<<liste[i]<<"\t\t Quantiter :
"<<qnt<<endl;
            qnt = qnt * prix[i];
            cout<<"Le prix : "<<qnt;

        }
    }while(cod!=Code[i] && i<=X);
}
```

```
}
```

Exercice 78:

```
#include <iostream> /****** On inclure la bibliotheque des entés/ sortie *****/
#include <conio.h>
#include <math.h>
#include <cstdlib> /*On inclure la bibliotheque qui sert principalement à déclarer la fonction
C system()*/
using namespace std; //espace de nom
struct point{ // Declaration de la structure
    float abs;
    float ord;
};
int main(){
    point A; // Declation de variable A de type point
    point B; // Declation de variable B de type point
    float d;
    cout<<"Veulliez saisir les coordonner de A: "<<endl;
    cout<<"abs= ";
    cin>>A.abs;
    cout<<"ord= ";
    cin>>A.ord;
    cout<<endl;
    cout<<"Veulliez saisir les coordonner de B: "<<endl;
    cout<<"abs= ";
    cin>>B.abs;
    cout<<"ord= ";
    cin>>B.ord;
    //On calcul la distance entre 2 points A et B
    d=sqrt(((B.abs-A.abs)*(B.abs-A.abs))+((B.ord-A.ord)*(B.ord-A.ord)));
    cout<<"la distance entre les deux point A et B est = "<<d<<endl;
    d=d/2;
    cout<<"le milieu des deux points A et B est = "<<d<<endl;
    system("pause");
    return 0;
}
```

Exercice 79:

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int i, j, k, nombre, cube, n,m;
    cout<<" Entrer un entier : ";
    cin>>n;

    if( n < 10)

        for(m = 1 ; m<=n ;m++)
            cout<<m<<endl;
```

```
else
{
    for( m=1; m<= 9;m++)
        cout<<m<<endl;
    for (i = 1 ;i<=n; i++)
    {
        for (j = 0; j<n;j++)
        {
            for (k = 0 ;k<n;k++)
            {
                nombre = 100 * i + 10 * j + k;
                cube = i * i * i + j * j * j + k * k * k;
                if (cube == nombre && cube<=n)
                {
                    cout<<nombre<<endl;
                }
            }
        }
    }
}

system("PAUSE");
}
```

Exercice 80:

/* On veut définir si un nombre saisi par l'utilisateur est Armstrong ou non.*/

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    int i, j, k, nombre, cube, n,m;
```

```
    cout<<" Entrez un entier ";
```

```
    cin>>n;
```

```
    bool trouve=false;
```

```
    if( n < 10)
```

```
        trouve=true;
```

```
    else
```

```
    {
```

```
        for (i = 1 ;i<=n; i++)
```

```
        {
```

```
            for (j = 0; j<n;j++)
```

```
            {
```

```
                for (k = 0 ;k<n;k++)
```

```
                {
```

```
                    cube = i * i * i + j * j * j + k * k * k;
```

```
                    if ( cube == n)
```

```
                    {
```

```
                        trouve=true;
```

```
                    }
```

```
    }  
  }  
}  
  
if(trouve==true)  
  cout<<"le nombre est Amstrong"<<endl;  
else  
  cout<<"le nombre n'est pas Amstrong"<<endl;  
  
system("PAUSE");  
  
}
```

1.3 Correction en langage .NET

Visual Basic .Net est un langage de programmation à la syntaxe similaire à celle de Visual Basic 6.

Néanmoins, ces deux langages sont assez peu comparables dans la pratique tant l'évolution entre ceux-ci est énorme. Le principal changement étant sans conteste l'introduction de l'orientation objet dans le langage.

VB.NET permet de développer en .Net via Visual Studio, c'est-à-dire seulement sur les systèmes d'exploitation Windows (98, 2000, XP, Vista, 7).

Syntaxe du VB.Net en Générale

| <i>Les instructions</i> | <i>Exemples</i> |
|--------------------------------------|--|
| Déclaration | Dim A As Integer |
| Saisie | Console.ReadLine() Console.In.ReadLine() |
| Afficher | Console.Out.WriteLine(("ligne à afficher")) Console.WriteLine ("Résultat") |
| Déclaration ,initialisation | Dim a As Integer = 10 'Déclare du variable nommée 'a' de type entier |
| if...then(Si...Alors) | If Condition 1 Then Action 1 Else Action 2 If Condition 2 Then Action 3 End If End If |
| Select Case (Selon le choix) | Select Case expression Case valeur1 |

Programmation structurée

| | |
|--|---|
| | <pre>'code effectué si expression=valeur1 Case valeur2 'code effectué si expression=valeur2 Case valeur3 'code effectué si expression=valeur3 .. Case Else 'code effectué dans tous les autres cas End Select</pre> |
| Logique (ou ,et ,non) | <pre>If B>2 And B<7 Then If B>2 Or B<7 Then If Not (B>2) Then</pre> |
| For ...next (pour) | <pre>For variable=début To fin Action répéter Next variable</pre> |
| Do Loop (faire jusqu'à) | <pre>Do Until condition Code Loop 'Boucler jusqu'à ce que condition soit vraie</pre> |
| Do While/LoopWhile (faire tant que) | <pre>Do While Condition Code Loop Ou Do Code Loop While Condition</pre> |
| While End While(tant que) | <pre>While Condition Code répéter End While</pre> |
| Les tableaux | <pre>Dim A(4)As Integer 'on déclare le tableau A(2)=4 x = A(2)</pre> |
| Les Structures | <pre>Public Structure Adresse Dim Numero As Integer Dim Rue As String Dim Ville As String End Structure</pre> |
| Procédure | <pre>Sub MaProcédure() Code End Sub</pre> |
| fonction | <pre>Function MaFonction() As Integer 'MaFonction qui retourne un entier par exemple Code Return Valeur</pre> |

Programmation structurée

| | |
|------------------------------|---|
| | End Function |
| Les opérateurs : | |
| Addition : + | Dim A,B, C As Integer B=2 C=3 A=B+C |
| Soustraction : - | Dim A,B, C As Integer B=2 C=3 A=B-C |
| Multiplication : * | Dim A,B, C As Integer B=2 C=3 A=B*C |
| Division : / | Dim A,B, C As Single B=2 C=3 A=B/C |
| Puissance : ^ | A=B^3 'A=B*B*B |
| Modulo : Mod | If A Mod 3 = 0 then... |
| Concaténation : & | A= "VISUAL" B= " " C= "BASIC" D=A & B & C donne D="VISUAL BASIC" |

Exercice 1 :

```
'Ce programme permet de :  
  '1)saisir deux entiers  
  '2)calculer la somme , la difference, la multiplication, et la division  
des deux entiers  
  '3)afficher les resultat des quatre operation
```

```
Sub Main()  
  
  Dim var1, var2 As Integer  
  
  Console.WriteLine("Veuillez saisir le 1ere entier SVP")  
  var1 = Console.ReadLine()  
  Console.WriteLine("Veuilles saisir la 2eme entier SVP")  
  var2 = Console.ReadLine()  
  Console.WriteLine("La somme des entiers est : ")  
  Console.WriteLine("La difference entre les entiers est : " & var1 -  
var2)  
  Console.WriteLine("La multiplication des entiers est : " & var1 *  
var2)  
  Console.WriteLine("La division des entiers est : " & var1 / var2)  
  Console.ReadLine()
```

End Sub

Exercice 2 :

```
'Ce programme permet de faire la permutation cyclique dans le sens
d'une éguit de mantre
'Soit A,B,C,D 4 variables réel
'il faut donc remplir les 4 variables , on retien la valeur de la 2eme
variable dans une autre nouvelle variable , ensuite , on retien faire
'la 1ere valeur dans la 2eme variable , la 2eme dans la 3eme , la 3eme
dans le 4eme , la valeur du nouvelle variable dans la 1ere - C'est ce
'que l'on appelé la permutation - , en fin , on affiche le resultat
Sub Main()
    Dim A, B, C, D, var1, var2 As Single
    Console.WriteLine("Veuillez saisir la valeur de A SVP")
    A = Console.ReadLine()
    Console.WriteLine("Veuillez saisir la valeur de B SVP")
    B = Console.ReadLine()
    Console.WriteLine("Veuillez saisir la valeur de C SVP")
    C = Console.ReadLine()
    Console.WriteLine("Veuillez saisir la valeur de D SVP")
    D = Console.ReadLine()

    var1 = B
    B = A
    Console.WriteLine("la valeur de B est : " & B)
    var2 = C
    C = var1
    Console.WriteLine("La valeur de C est : " & C)
    var1 = D
    D = var2
    Console.WriteLine("La valeur de D est : " & D)
    A = var1
    Console.WriteLine("La valeur de A est : " & A)
```

End Sub

Exercice 3

```
'Ce programme permet de :
'- saisir 2 entiers
'- comparer les deux valeurs des variables
'- afficher le plus gran d'entre eux
' + si les valeurs des variables son egaux , on affiche un message de
l'égalité
```

```
Sub Main()
    Dim var1, var2 As Integer 'Déclaration de deux variables

    Console.WriteLine("Veuillez saisir le 1ere entier SVP") 'affichage
de saisir
    var1 = Console.ReadLine() 'lecture de l'écriture
et placé dans la variable n:1
    Console.WriteLine("Veuillez saisir la 2eme entier SVP") 'affichage
de saisir
    var2 = Console.ReadLine() 'lecture de l'écriture et placé dans la
variable n:2
    If (var1 < var2) Then 'comparéson entre les 2 varibales
```

```
        Console.WriteLine("le plus grand entre les deux entier est :" &
var2) 'afficher le plus grand des entiers

        Else ' si la valeur du 1er variable n'est pas inferieur de l'autre
on comparr les dans les autres cas possible
            If (var1 > var2) Then 'comparéson entre les 2 varibales

                Console.WriteLine("le plus grand entre les deux entier est :"
& var1) 'afficher le plus grand des entiers

                    Else 'si la valeur de l'un est ni grand ni petite que l autre :
alors les valeurs son egaux

                        Console.WriteLine("les deux entiers sont egaux")
'//afficher que les entiers saisie est egaux
                            End If
                                End If

                                    End Sub
```

Exercice 4

```
' Le programme qui permet de :
'- saisir 3 entiers
'- comparer les trois valeurs des variables
'- afficher le plus gran d'entre eux
' + si les valeurs des variables son egaux , on affiche un message de
l'égalité
Sub Main()
    Dim var1, var2, var3, max As Integer ' déclaration des variable de
type entier
    Dim egaux As Boolean 'déclaration d un variable de type booleen

    Console.WriteLine("Veuillez saisir un entiers SVP") 'affichage de
saisir

    var1 = Console.ReadLine 'lecture de l'écreture et placé dans la
variable n:1

    Console.WriteLine("Veuillez saisir un entiers SVP") 'affichage de
saisir

    var2 = Console.ReadLine 'lecture de l'écreture et placé dans la
variable n:2

    Console.WriteLine("Veuillez saisir un entiers SVP") 'affichage de
saisir

    var3 = Console.ReadLine 'lecture de l'écreture et placé dans la
variable n:3

    If (var1 > var2) Then 'tester si var1 est inferieur a var2

        max = var1 'affectation du var1 au varibale max

    Else

        If (var2 > var1) Then 'tester si var2 est inferieur a var1
```

Programmation structurée

```
        max = var2 'affectation du var1 au varibale max
    Else
        max = var1 'affectation du var2 au varibale max
        egaux = True 'on retien la valeur vrai pour la variable
egaux
    End If
End If

If (max > var3) Then 'tester si max est inferieur a var3
    Console.WriteLine("le plus grand est : " & max) ' l affichage
du plus grand des entiers
Else
    If (max < var3) Then 'tester si max est supperieur a var3
        Console.WriteLine("le plus grand est : " & var3) 'l
affichage du plus grand des entiers
    Else 'sinon
        If (egaux = False) Then 'tester si la valeur de la variable
egaux est vrai
            Console.WriteLine("les 3 entiers sont egaux ") '
afficher que le 3 entiers son egaux
        Else 'sinon
            Console.WriteLine("le 1ere et la 3eme entier sont egaux
" & var1) 'affichage que 2 entier sont egaux
            End If
        End If
    End If

End Sub
```

Exercice 5

```
'le resoudre de l'equation  $ax+b=0$  ,a besoin d'un programme qui permet de
:
'          + Saisir deux valeurs ( le 1er est le coeficient du X ; et
la 2eme c'est le terme independante)
'          + tester le 1er valeur :
'          - si il est differente a 0 ( on fait 1 operation
Appropriées a ce condition)
'          - sinon on faire un autre teste pour le terme
independante ,si est egale a 0 ( on fait 1 operation
'          Appropriées a ce condition)
'          -- sinon on affiche juste un message
'          + a chaque fois on affiche un message ça depend au saisi de
l'utilisateur

Sub Main()
    Dim a, b, c As Single 'déclaration des variables de type reel
```

```
        Console.WriteLine("          ax+b=0")

        Console.WriteLine("Veuillez SVP saisir la valeur de a : ") '
affichage qui demande a l'utilisateur de saisir la valeur de la coeficient
de X
        a = Console.ReadLine() 'lire le coeficient

        Console.WriteLine("Veuillez SVP saisir la valeur de b : ") '
affichage qui demande a l'utilisateur de saisir la valeur du terme
independante
        b = Console.ReadLine() 'lire le terme independante

        If (a! = 0 And b! = 0) Then ' on teste si la valeur de a est
different a 0 ( != c'est le signe different en c++

            c = -b / a ' faite la calcule

            Console.WriteLine("la solution est : " & c) 'on affiche la
solution

        Else

            If (b = 0) Then 'on teste si la valeur de b est egale a 0

                Console.WriteLine("Toute possibilite") 'on affiche un
message du resultat

            Else

                Console.WriteLine("Pas de solution") 'on affiche un message
du resultat

            End If

        End If
    End Sub
```

Exercice 6

'le resoudre de l'equation $ax^2+bx+c=0$,a besoin d'un programme qui permet de :

```
'
'      + Saisir trois valeurs ( le 1er est le coeficient du X2 , le
2eme est le coeficient du X; et la 3eme est le
'      terme independante)
'
'      + tester le 1er valeur
'      - si il est differente a 0 ( on fait 1 operation
Appropriées a l'equation du 1ere degre)
'      - sinon on faire la calcule de delta
'      -- si delta est superieur a 0, donc on calcule le X1 et le
X2( le meme methode que le math (-b-[racin de delta]sur 2*a)).
'      -- si delta est inferieur a 0, on affiche un message
'      -- si delta st egal a 0 , on calcule juste X1
'
'      + a chaque fois on affiche un message ça depend au saisi de
l'utilisateur
```

```
Sub Main()

    Dim a, b, c As Integer 'déclaration des varibales de type entier

    Dim X1, X2, delta As Single 'déclaration des varibales de type reel

    Console.WriteLine("Veuillez saisir le coeficient du X2 : ")
'affichage pour le saisie
    a = Console.ReadLine() 'lecture de la saisie du coeficient du X2

    Console.WriteLine("Veuillez saisir le coeficient du X : ")
'affichage pour le saisie
    b = Console.ReadLine() 'lecture de la saisie du coeficient du X

    Console.WriteLine("Veuillez saisir le term independant : ")
'affichage pour le saisie
    c = Console.ReadLine() 'lecture de la saisie du terme independant

    If (a = 0) Then 'on teste si la valeur du a est egale a 0

        X1 = -c / b 'on fait la calcule
        Console.WriteLine("Equation du 1ere degre.") 'simple affichage
        Console.WriteLine("la valeur de x: " & X1) 'afficher la valeur
de X1
    Else

        Console.WriteLine("delta =(b*b)-4*c*a")
        delta = (b * b) - (4 * c * a) 'on fait la calcule du delta

        If (delta > 0) Then 'on teste si la valeur de delta est
inferieur a 0

            Console.WriteLine("la valeur de delta >0") 'afficher la
valeur du delta
            X1 = (-b - Math.Sqrt(delta)) / (2 * a) 'calculer X1

            Console.WriteLine(" X1 = " & X1) 'afficher la valeur du X1
            X2 = (-b + Math.Sqrt(delta)) / (2 * a) 'calculer X2
            Console.WriteLine(" X2 = " & X2) 'afficher la valeur du X2

        Else

            If (delta < 0) Then 'on teste si la valeur de delta est
superieur a 0

                Console.WriteLine("l'ensemble des solutions vide")
'affichage du message

            Else
                X1 = -b / (2 * a) 'claculer X1
                Console.WriteLine("La valeur de X1 : " & X1) 'afficher
le X1
            End If
        End If
    End If
End Sub
```

Exercice 7

```
'le programme suivante faire le calcule du manton de la participation
au pris du repas de ses emplois de la façon suivanate:
'      +si l'employer est cilibataire la particiation est de 20%
```

Programmation structurée

```
'
'      +si l'employeur est marier la paricipation est de 25%
'      -- si il a des enfants la participation est de 10%
'      +-+ la participation est plafonné a 50%
'      --si le salaire mensuele est inferieur a 6000 Dhs la
participation est majoré de 10%
'
'      +a chaque fois quand a le resultat de la participation on
affiche le

Sub Main()
    Dim nom, sit As String 'declaration des variables de type chaine de
caractaire
    Dim slr, NBENF As Integer 'declaration des variables de type entier
    Dim prix, prtcp As Single 'declaration des variables de type reel

    Console.WriteLine("veuillez saisir votre nom SVP ") 'affichage
pour le saisie
    nom = Console.ReadLine() ' lecture du saisie

    Console.WriteLine("veuillez saisir votre salaire SVP(DHs) ")
'affichage pour le saisie
    slr = Console.ReadLine() 'la lecture du saisie

    Console.WriteLine("veuillez saisir le prix de ropas SVP(DHs) ")
'affichage pour le saisie
    prix = Console.ReadLine() 'la lecture du saisie

    Console.WriteLine("veuillez saisir votre situation familiale SVP
") 'affichage pour le saisie
    sit = Console.ReadLine() 'la lecture du saisie

    If (slr > 6000) Then 'tester si le salaire est superieur a 6000
dhs

        If (sit = "cilibataire") Then 'tester si la situation du
employier est clibataire

            prtcp = (prix * 20 / 100) 'calculer la participation
            Console.WriteLine("votre participation M." << nom << "
est : " << prtcp << " DH ") 'affichage du participation

        Else

            Console.WriteLine("vauillez saisir le nombre des enfants
svp :)") 'affichage du saisie
            NBENF = Console.ReadLine() 'la lecture de saisie

            If (NBENF = 0) Then 'tester si l employeur a des enfants

                prtcp = (prix * 25 / 100) 'calculer la participation
                Console.WriteLine("votre participation M." << nom << "
est : " << prtcp << " DH ") 'affichage du participation

            Else
```

Programmation structurée

```
        If (NBENF >= 3) Then 'tester si le nombre des enfants
est superieur a 3

            prtcp = (prix * 50 / 100) 'calculer la
participation
            Console.WriteLine("votre participation M." << nom
<< "  est : " << prtcp << " DH ") 'affichage du participation

            Else

                prtcp = (prix * 25 / 100 + prix * NBENF * 10 / 100)
'calculer la participation
                Console.WriteLine("votre participation M." << nom
<< "  est : " << prtcp << " DH ") 'affichage du participation
                End If
            End If
        End If

else
        If (sit = "cilibataire") Then 'tester si la situation du
employeur est cilibataire

            prtcp = (prix * 20 / 100) 'calculer la participation
prtcp = (prtcp + prix * 10 / 100) 'on rajoute la
participation effectuer
            Console.WriteLine("votre participation M." & nom & "  est
: " & prtcp & " DH ") 'affichage du participation
            Else

                Console.WriteLine("veuillez saisir le nombre des enfans
svp :)") 'affichage du saisie
                NBENF = Console.ReadLine() 'la lecture du saisie

                If (NBENF = 0) Then 'tester si il a des enfants

                    prtcp = (prix * 25 / 100) 'calculer la participation
                    prtcp = (prtcp + prix * 10 / 100) 'on rajoute la
participation effectuer
                    Console.WriteLine("votre participation M." & nom & "
est : " & prtcp & " DH ") 'affichage du participation

                    Else
                        If (NBENF >= 3) Then 'tester si le nombre d enfants
superieur a 3

                            prtcp = (prix * 50 / 100) 'calculer la
participation
                            prtcp = (prtcp + prix * 10 / 100) 'on rajoute la
participation effectuer
                            Console.WriteLine("votre participation M." & nom &
"  est : " & prtcp & " DH ") 'affichage du participation

                            Else

                                prtcp = (prix * 25 / 100 + prix * NBENF * 10 / 100)
'calculer la participation
                                prtcp = (prtcp + prix * 10 / 100) 'on rajoute la
participation effectuer
                                Console.WriteLine("votre participation M." & nom &
"  est : " & prtcp & " DH ") 'affichage du participation
```

```
        End If
    End If
End If
```

```
End Sub
```

Exercice 8

' ce programme faite pour que soit un étudiant ayont cinq notes dans l'examen.

'donc on faite la calcule de la moyenne de cette etudiant sachant que toutes les notres on le meme coeficients

'alors on demanade a l utilisateur d'entrer les notes pour que calcculer la moyenne general et on affichent le resultat

```
Sub Main()
    Dim i As Integer ' declaration d un entier
    Dim note, somme As Single 'declaration des variables de types reel

    somme = 0 'inistatisation par 0
    For i = 0 To 4 'la boucle pour

        Console.WriteLine("Veillez saisir une note SVP : ") 'affichage
pour faire la saisi des notes
        note = Console.ReadLine() 'la lecture des notes
        somme = note + somme 'calculer la somme des notes

    Next

    note = somme / 5 'calculer la moyenne des notes
    Console.WriteLine("la moyene est : " & note) 'afficher la moyenne
general
End Sub
```

Exercice 9

' ce programme est pour foction pricipale c'est de donnée une table de multiplication jusqu'a 10

'prmierement , la calcule de la multiplication de l entier a saisit (on demande dès le debut de saisir un entier) avec le " i" qui

'est changer a chaque fois avec le pas +1 jusqu'a le chiffre 10

' enfin, l'affichage du resultat de chaque multiplication

' (donc le resultat , une table de multiplication d'un chiffre saisite).

```
Sub Main()
    Dim multi, var, i As Integer 'declaration des svariables de types
entier
    Console.WriteLine("Veillez saisir un entier : ") 'affichage pour la
saisi d une variable par l utilisateur
    var = Console.ReadLine() ' lecture du variable
    For i = 1 To 10 'boucle pour

        multi = var * i 'calculer la multiplication
```

```
        Console.WriteLine(var & " * " & i & " = " & multi) 'affichage
de la table de multiplication
    Next
End Sub
```

Exercice 10

```
'le programme suivante permet de :
'
'      + calculer la somme de la suite S=1+2+3+.....+n
'      - on calcule la somme des chiffres jusqu'a l'element saisi
par l'utilisateur
'
'      + afficher le resultat de la somme de la suite

Sub Main()
    Dim somme, suite, i As Integer ' des variables de type entier

    somme = 0 'initialisation de la variable somme par 0

    Console.WriteLine("veuillez saisir le nombre de la suite : ")
'affichage de saisie
    suite = Console.ReadLine() 'la lecture de saisie

    For i = 0 To suite 'boucle pour

        somme = somme + i ' calculer la somme de la suite
    Next

    Console.WriteLine("La somme de la suite " & suite & " est : " &
somme) 'afficher le resultat de la suite

End Sub
```

Exercice 11

```
'le programme suivante permet de :
'
'      + calculer la somme de la suite S=1-2+3-4.....(+-)n
'      - donc la suite contient des chiffres pairs et autres
impaires, c'est pour cela on fait le calcul de la somme
'      des chiffres pairs et aussi la somme des chiffres impaires
jusqu'a l'element saisi par l'utilisateur
'      - après on fait la différence entre la somme des chiffres pairs
et impaires
'
'      + afficher le resultat de la suite

Sub Main()
    Dim n, i, somme1, somme2 As Integer 'declaration des variables de
type entier

    somme1 = 0 'initialisation de la somme 1 par 0
    somme2 = 0 'initialisation de la somme 2 par 0

    Console.WriteLine("veuillez saisir un entier SVP : ") 'affichage de
saisie
```

```
n = Console.ReadLine() 'la lecture de saisie

For i = 1 To n Step 2 'boucle pour

    somme1 = somme1 + i 'calculer la somme des entier paire
    somme2 = somme2 + (i + 1) 'calculer la somme des entier impaire
Next
Console.WriteLine(" la somme est : " & somme1 - somme2) 'calculer
la difference entre la somme des paire et la somme des impaire et après on
affiche le resultat

End Sub
```

Exercice 12

```
'le factoriel d'un entier N!= N*(N-1)*(N-2)*.....*1
'
'le programme ont pour fonction pricipale de faire le calcule de cette
factoriel N!
'
' donc on demande a l'utilisateur de saisir un entier pour faire le
calcule ;
'on travail sur ce entier ( :la multiplication des nombres a partir de
'1' jusqu'a l'element saisie par l'utilisateur);
'et enfin on affiche le resultat
Sub Main()
    Dim N, Var1, i As Integer 'declaration des variables de type entier
    Var1 = 1 'inisialisation du variable Var1 par 1
    Console.WriteLine("veuillez saisir un entier SVP ") 'affichage de
saisie
    N = Console.ReadLine() 'lecture de saisie
    For i = 1 To N 'boucle pour
        Var1 = i * Var1 'calculer la factoriel du N qui est saisite par
l utilisateur
    Next
    Console.WriteLine("le factoriel " & N & "! est: " & Var1) 'afficher
le factoriel du N
End Sub
```

Exercice 13

```
' ce programme faite pour que soit un étudiant ayont X notes dans
l'examen.(nombre de notes de l'etudiant n'est pas connu)

'donc on faite la calcule de la moyenne de cette etudiant,
'sachant que toutes les notes ont des coeficients differents ;
'alors on demanade a l utilisateur d'entrer les notes pour que
'calculer la moyenne general ( pour ça on est besoin de calculer
'la somme des coeficients a part et ainsi la somme des notes multiplier
par leurs coeficients )
'a la fin , on affichent le resultat de la moyenne

Sub Main()
    Dim coef, somme1, i As Integer 'declaration des variables de type
entier
    Dim Note, N, somme2, res As Single 'declaration des variables de
type reel

    Console.WriteLine("veuillez saisir le nombre des notes SVP : ")
'affichage de saisie du nombre des notes pour gere l arret de les boucles
suivantes
```

```
N = Console.ReadLine() ' lecture de N

somme1 = 0 'inialization de la variables somme1 par 0
somme2 = 0 'inialization de la variables somme2 par 0
i = 1 'inialization de la variables i par 1

While (i <= N) 'la boucle Tant que

    Console.WriteLine("Veuillez saisir une note : ") 'affichage de
saisie d une note
    Note = Console.ReadLine() 'la lecture du note
    Console.WriteLine("Veuillez saisir le coeficient de cette note
: ") 'affichage de saisie d un coeficient de cette note
    coef = Console.ReadLine() 'la lecture du coeficient
    somme1 = somme1 + coef 'calculer la somme des coeficients
    somme2 = somme2 + (coef * Note) 'calculer la somme des
coeficients multiplier par la note
    i = i + 1 'gere le compteur
End While

    res = somme2 / somme1 'calculer la moyenne des notes
    Console.WriteLine("La moyenne est : " & res) 'afficher le resultat
( la moyenne des notes)
End Sub
```

Exercice 14

```
' ce programme affiche le maximum , le minimum , la somme , et la
moyenne des entiers saisis par l utilisateur
' Au debut , on demande a l'utilisateur de saisir un entier pour que on
debut la comparaison avec les entiers suivantes
', ensuite on teste si la valeur suivante est plus grand que le
precedente , on prend le dans un variable qui correspondante ;
'sinon on teste si il est plus petit, et aussi en le prend dans le
variable qui correspondante
' A chaque fois ,on faite la somme des entiers , enfin on calcule la
moyenne de ces entiers et on l'affiche le resultat
```

```
Sub Main()
    Dim var1, max, min, j, i, somme As Integer 'Déclaration des
variables de type entier
    Dim moy As Single 'declaration d un variable de type reel
    Dim rep As String 'declaration d'un variable de type chaine de
caractere

    i = 1 'initialisation du compteur par un 1

    Console.WriteLine("element " & i & " : ") 'affichage de saisir
var1 = Console.ReadLine() 'lecture de l'écriture et placé dans la
variable var1

    max = var1 '
    min = var1 ' initialisation du (min , max, et somme par var1)
    somme = var1 '
Do

    Console.WriteLine("element " & i + 1 & " : ") 'affichage de
saisir
    var1 = Console.ReadLine() 'lecture du contenu de la variable
var1
```

```
    If (max < var1) Then 'comparéson entre les 2 varibales
        max = var1 'affectation du contenu du variable var1 au
variable max

    Else ' si la valeur du 1er variable n'est pas inferieur de
l'autre on comparr les dans les autres cas possible

        If (min > var1) Then 'comparéson entre les 2 varibales
            min = var1 'affectation du contenu du variable var1 au
variable min

        End If
    End If

    somme = somme + var1 'calculer la somme des entiers saisie
    i = i + 1 'Incrémenter le compteur

    Console.WriteLine("Voulez-vous continuer?(oui/non) ")
'demander a l'utilisateur de saisir "oui" si il veut resaisir un autre
entier
    rep = Console.ReadLine() 'la lecture de la reponse

    Loop While (rep = "oui")

    moy = somme / i - 1 'calculer la moyenne

    Console.WriteLine("le maximum des entiers est : " & max) '
affichage du resultat
    Console.WriteLine("le minimum des entiers est : " & min) '
affichage du resultat
    Console.WriteLine("la somme des entiers est : " & somme) '
affichage du resultat
    Console.WriteLine("la moyenne est : " & moy) '
affichage du resultat

End Sub
```

Exercice 15

'On desir calculer le montant de la facture d'électricité d'un abonner dans ce programme de la façon suivante :

```
' + La consommation selon un tarif à tranche :
' - 0.9 Dh/KW pour les 110 premier KW
' - 0.98 Dh/KW pour les 110 KW suivant
' - 1.20 Dh/KW pour les KW supperieur a 220 KW
'
' * le programme permet de :
' + demander a l'utilisateur d'entrer le code
' + demander a l'utilisateur d'entrer le nom
' + demander a l'utilisateur d'entrer le total de la consommation
d'électriciter par mois
' + Calculer le montant Hortaxt par tranche
' + Calculer le montant TTC par tranche
' + Calculer le montant Totale de la facture plus les frais fixe
d'abonnement : 70 Dh
```

```
' (si la consommation est inferieur de 110 , on fait la calcule du
montant hortaxte,
' , TTC et totale pour une seul tranche , si la consommation est
superieur a 110 et inferieur
' a 220 ;on fait la calcule du montant hortaxte, TTC et totale pour
le prmeier et la deuxieme tranche ,
' mais quand la consommation est superieur de 220 , on fait la
calcule du montant hortaxte,
' , TTC et totale pour le premier , la deuxieme et la troisieme tranche
)

'+ A chaque fois ,on fait l'affichage du resultat avec le nom ,le code
et toutes les information

Sub Main()
    Dim conso, hort1, hort2, hort3, ttc1, ttc2, ttc3, tot As Single
'declaration des variables de types réel
    Dim code As Integer 'declaration d'une variable "Code" de type
entier
    Dim nom, rep As String 'declaration des variables de type chaine de
caractère

    Do 'faire

        Console.WriteLine("veuillez saisir votre Nom : ") 'affichage
pour saisir le nom
        nom = Console.ReadLine() 'la lecture de nom
        Console.WriteLine("veuillez saisir votre Code : ") 'affichage
pour saisir le code
        code = Console.ReadLine() 'la lecture de code
        Console.WriteLine("veuillez saisir votre consommation : ")
'affichage pour saisir la consommation
        conso = Console.ReadLine() ' lecture de consommation

        If (conso <= 110) Then 'tester si la consommation est inferieur
de 110

            Console.WriteLine("nom :" & nom & "\t Code :" & code)
            Console.WriteLine(" la consommation :" & conso & " Dh")

            hort1 = conso * 0.9 'calculer le hortaxt
            Console.WriteLine("le mantont hortaxt est : " & hort1 & "
Dh") 'afficher le mantont du hortaxt
            ttc1 = hort1 + (hort1 * 0.2) 'calculer le TTC du ler
Tranche
            Console.WriteLine("le montant TTC est : " & ttc1 & " Dh")
'afficher le montant du TTC
            tot = ttc1 + 70 'calculer le totale de la facture
            Console.WriteLine("le totale de la facture est : " & tot &
" Dh") 'afficher le totale de la facture

        Else 'si la consommation est superieur de 110

            If (conso > 110 And conso < 220) Then 'tester si la
consommation est superieur a 110 et inferieur a 220
```

```
        Console.WriteLine("nom :" & nom & "\t Code :" & code)
        Console.WriteLine(" la consommation :" & conso & " Dh")
        hort1 = 110 * 0.9 'calculer le hortaxt du 1er Tranche
        Console.WriteLine("le 1er mantont hortaxt est : " &
hort1 & " Dh") 'afficher le mantont du 1er hortaxt
        hort2 = (conso - 110) * 0.98 'calculer le hortaxt du
2eme Tranche
        Console.WriteLine("le 2eme mantont hortaxt est : " &
hort2 & " Dh") 'afficher le mantont du 2eme hortaxt
        ttc1 = hort1 + (hort1 * 0.2) 'calculer le TTC du 1er
Tranche
        Console.WriteLine("le 1er montant TTC est : " & ttc1 &
" Dh") 'afficher le 1er montant du TTC
        ttc2 = hort2 + (hort2 * 0.2) 'calculer le TTC du 2eme
Tranche
        Console.WriteLine("le 2eme montant TTC est : " & ttc2
& " Dh") 'afficher le 2eme montant du TTC
        tot = ttc1 + ttc2 + 70 'calculer le totale de la
facture
        Console.WriteLine("le totale de la facture est : " &
tot & " Dh") 'afficher le totale de la facture

    Else 'si la consommation est superieur de 220

        Console.WriteLine("nom :" << nom << "\t Code :" &
code)
        Console.WriteLine(" la consommation :" & conso & " Dh")
        hort1 = 110 * 0.9 'calculer le hortaxt du 1er Tranche
        Console.WriteLine("le 1er mantont hortaxt est : " &
hort1 & " Dh") 'afficher le montant du 1er hortaxt
        hort2 = 110 * 0.98 'calculer le hortaxt du 2eme Tranche
        Console.WriteLine("le 2eme mantont hortaxt est : " &
hort2 & " Dh") 'afficher le montant du 2eme hortaxt
        hort3 = (conso - 220) * 1.2 'calculer le hortaxt du
3eme Tranche
        Console.WriteLine("le 2eme mantont hortaxt est : " &
hort3 & " Dh") 'afficher le montant du 3eme hortaxt
        ttc1 = hort1 + (hort1 * 0.2) 'calculer le montant du
1er TTC
        Console.WriteLine("le 1er montant TTC est : " & ttc1 &
" Dh") 'calculer le TTC du 1er Tranche
        ttc2 = hort2 + (hort2 * 0.2) 'calculer le montant du
2eme TTC
        Console.WriteLine("le 2eme montant TTC est : " & ttc2
& " Dh") 'calculer le TTC du 2eme Tranche
        ttc3 = hort3 + (hort3 * 0.2) 'calculer le montant du
3eme TTC
        Console.WriteLine("le 2eme montant TTC est : " & ttc3
& " Dh") 'calculer le TTC du 3eme Tranche
        tot = ttc1 + ttc2 + ttc3 + 70 'calculer le mantant
totale de la facture
        Console.WriteLine("le totale de la facture est : " &
tot & " Dh") 'afficher le totatle de la facture
    End If

End If

        Console.WriteLine("voulez vous continuez (oui/non)?")
'affichage pour saisir si il veux continuer taper oui sinon taper non
        rep = Console.ReadLine() 'lir ela reponse de saisie
```

```
Loop While (rep = "oui") 'tester si la reponse est existe
End Sub
```

Exercice 16

```
'ce programme permet de :
'      + saisir le nom et le nom de l'utilisateur
'      + saisir le jour, le mois , l'annee de naissance de
'utilisateur
'      + saisir le jour, le mois , l'annee actuelle

'* A chque fois quand l'utilisateur a saisie une date , on doit faire un
contrôle de validité
'(si une date n'est pas valides on affiche un message d'erreur a
l'utilisateur , et on donnee a lui
'la possibiliter de resaisir la date )
'
'- le jour doivent être compris entre 1 et 31
'- le mois doivent être compris entre 1 et 12
'- l'annee doit être obligatoirement inferieur ou égale a 2010

'* Si les valeurs sont valide , on fait le calcule l'âge en nombre
d'année entiere

'* on affiche un message de resultat de l'âge ( si le jour et le mois
de naissance
'correcpandent au jour et mois actuelle on afiche un message de joyeux
anniverssaire )

Sub Main()
Dim sortire As Boolean 'declaration d une variable de type booleen
Dim jourN, jourA, moisA, moisN, AnneN, anneeA, age As Integer
'declaration des variables de type entier
Dim rep, nom, prenom As String ' declaration des variables de type
chaîne de caractaire

rep = "non" 'inialisisation du variable "rep" par la valeur non
sortire = False 'intialisisation du variable "sortire " par la valeur
faux

Do ' faire N°1

Console.Write("Nom : ") 'affichage pour saisie du nom
nom = Console.ReadLine() 'lecture du nom
Console.Write("Prenom : ") 'affichage pour saisie du prenom
prenom = Console.ReadLine() 'lecture du prenom

Do ' faire N°2

Console.Write("Jour de naissance : ") 'affichage pour
saisire le jour de naissance
jourN = Console.ReadLine() 'lire le jour de naissance

If (jourN < 1 Or jourN > 31) Then ' Si N° 1 *** tester si
le jour de naissance est superieur a 1 et inferrieur a 31
```

Programmation structurée

```
        Console.WriteLine("Erreur (La date du jour est invalide) .  
Voulez-vous recommencez (oui/non)?") 'afficher un erreur si la date qui  
l'utilisateur a saisie est invalide  
        rep = Console.ReadLine() 'lire la reponse de resaisire  
  
    Else 'SINON N°1  
  
        Do 'faire N°3  
  
            Console.WriteLine("Mois de naissance : ") 'affichage  
pour saisir le mois de naissance  
            moisN = Console.ReadLine() 'lire le mois de  
naissance  
  
            If (moisN < 1 Or moisN > 12) Then 'Si N° 2  
  
                Console.WriteLine("Erreur (La date du mois est  
invalide) . Voulez-vous recommencez (oui/non)?") 'afficher un erreur si la  
date qui l'utilisateur a saisie est invalide  
                rep = Console.ReadLine() 'lire la reponse de  
resaisire  
  
            Else 'Sinon N° 2  
  
                Do 'faire N°4  
  
                    Console.WriteLine("Annee de naissance : ")  
'affichage pour saisir 1 annee de naissance  
                    AnneN = Console.ReadLine() 'lire 1 annee  
de naissance  
  
                    If (AnneN > 2010) Then 'si N°4  
  
                        Console.WriteLine("Erreur (La date d'annee  
est invalide) . Voulez-vous recommencez (oui/non)?") 'afficher un erreur si  
la date qui l'utilisateur a saisie est invalide  
                        rep = Console.ReadLine() 'lire la  
reponse de resaisire  
  
                    Else 'sinon N°4  
  
                        Do 'faire N°5  
  
                            Console.WriteLine("Jour Actuelle : ")  
'affichage pour saisie le jour actuelle  
                            jourA = Console.ReadLine() 'lire le  
jour actuelle  
  
                            If (jourA < 1 Or jourA > 31) Then  
  
                                Console.WriteLine("Erreur (La date  
du jour est invalide) . Voulez-vous recommencez (oui/non)?") 'afficher un  
erreur si la date qui l'utilisateur a saisie est invalide  
                                rep = Console.ReadLine() 'lire  
la reponse de resaisire  
  
                            Else  
  
                                Do 'faire N°6
```

Programmation structurée

```
Console.WriteLine("Mois
actuelle : ") 'affichage pour saisie le mois actuelle
moisA = Console.ReadLine()
'lire le mois actulle
If (moisA < 1 Or moisA >
12) Then 'SI N°6
Console.WriteLine("Erreur
(La date du jour est invalide) . Voulez-vous recomencez (oui/non)?")
'afficher un erreur si la date qui l'utilisateur a saisie est invalide
rep =
Console.ReadLine() 'lire la reponse de resaisire
Else
Do 'faire N°7
Console.WriteLine("Annee actuelle : ") 'affichage pour saisie l annee actuelle
anneeA =
Console.ReadLine() 'lire l annee actuelle
If (anneeA <= 2009)
Then 'Si N°7
Console.WriteLine("Erreur (La date du jour est invalide) . Voulez-vous
recomencez (oui/non)?") 'affichage un erreur si la date qui l'utilisateur a
saisie est invalide
rep =
Console.ReadLine() 'lire la reponse de resaisire
Else
age = anneeA -
AnneN 'calculer la difference entre l'annee actuelle et l'annee de
naissance
Console.WriteLine("Bonjour " & nom & " " & prenom & " vous avez " & age &
" ans.") 'affichage le message avec un mon , prenom , et age de l
utilisateur
If (jourA =
jourN And moisA = moisN) Then 'si n:8 *****
Console.WriteLine("*****Joyeux
anniverssaire*****") 'affichage un message pour dire a l
utilisateur Joyeux anniversaire
End If ' si n:8
rep = "oui"
sortire = True
End If ' SI ET SINN
N°7
Loop While (rep = "oui"
And sortire = False) 'tant que N°7
```

```

End If ' si et SINN N°6
Loop While (rep = "oui" And
sortire = False) 'tant que N°6
End If ' Si et Sinon N°5
Loop While (rep = "oui" And sortire =
False) 'Tant Que N°5
End If ' si et sinon N°4
Loop While (rep = "oui" And sortire = False)
'fin du tant que N°4
End If ' Si et Sinn N°3
Loop While (rep = "oui" And sortire = False) 'fin tant
que N°3
End If ' de si et sinon N° 1
Loop While (rep = "oui" And sortire = False) 'fin de faire Tant
Que N° 2
Console.WriteLine("voulez-vous continuez (oui/non)?")
'affichage pour resaisire autres informations avec une valeur : "oui" ou
bien "non"
rep = Console.ReadLine() 'lire la reponse de saisie
Loop While (rep = "oui") 'fin de faire Tant Que N° 1
End Sub
```

Exercice 17

```
'ce programme permet de lire une suite N entiers dans un tableau
'd'abord, on donne la possibiliter a l utilisateur de saisir des
entiers pour remplir le tableau
'dans ce tableau on doit lire un nombre entier comme donnée et chercher
si se nombre existe
'
'* si existe on affiche un message
'* si n'existe pas on affiche un autre message

Sub Main()
Dim Tent(10) As Integer ';//declaration d un tableau qui est
nomer par " Tent " et on donne la taille du tableau : 10 case
Dim i, entier As Integer ' ;//declaration des variables de type
entier
Dim trouve As Boolean ';//declarer une variable "trouve" de type
boolean
trouve = False ';//inisialisation du variable "trouve" par faux
For i = 0 To 9 'la boucle pour
Console.WriteLine("Veuillez saisir un entier SVP : ") '
affichage de saisie pour remplire le tableau
```

Programmation structurée

```
Tent(i) = Console.ReadLine() '//lire les variables et mettent
chaque variable dans un case

Next

Console.WriteLine("veuillez entrer un entier SVP: ") '//afficher
de saisie un entier
entier = Console.ReadLine() 'lire

i = 0
Do

    If (Tent(i) = entier) Then 'tester si l'entier saisie est dans
le tableau

        Console.WriteLine("L'entier est existe : ") '/affichage
pour dire que l'entier est existe deja dansle tableau
trouve = True

        i = i + 1

    End If
Loop While (i <> 0 And trouve = False)

If (trouve = False) Then
    Console.WriteLine("l'element n'existe pas") 'affichage pour
dire que l'entier n'est existe pas dans le tableau
End If
End Sub
```

Exercice 18

```
' dans un tableau on range des entiers saisite par l'utilisateur(
maximum 50 entiers)

'ce programme faite pour afficher le maximum , le minimum , et la
moyenne des entiers a été saisie.

'il doit permettre de contrôler la saisie .

'(si l'utilisateur souhaite saisir moins de 50 entiers il saisi le
chiffre 0)

'pluse que ça , quand le minimum ou le maximum est trouver on prend la
position de chacun et on l'affiche

Sub Main()
    Dim i, j, min, max, moy, posmin, posmax As Integer 'declaration des
variables de type entier
    Dim Tab(10) As Integer ' declaration d un tableau de type entier
avec une taille de 10

    posmin = 0 ' inisialisation par 0
    posmax = 0 ' inisialisation par 0
    i = 0 ' inisialisation par 0

Do
```

```
        Console.WriteLine("Element" & i + 1 & " : ") '
        Tab(i) = Console.ReadLine() '
Remplire le tableau
        i = i + 1 '
Loop While (Tab(i - 1) <> 0 And i <> 10) '

If (i = 10) Then ' tester si la valeur de i est egale a 10

        i = i + 1
End If

max = Tab(0) 'Initialisation du Min, Max et moy par le 1er valeur
saisit
min = Tab(0) 'Initialisation du Min, Max et moy par le 1er valeur
saisit
moy = Tab(0) 'Initialisation du Min, Max et moy par le 1er valeur
saisit

For j = 1 To i - 2

        If (min > Tab(j)) Then 'tester si la nouvelle valeur est plus
petit que la precedante

                min = Tab(j) '//affectation de la valeur de Tab[j] au
variable min
                posmin = j '//affectation pour prendre la position du min

        Else

                If (max < Tab(j)) Then 'tester si la nouvelle valeur est
plus grand que la precedante

                        max = Tab(j) 'affectation de la valeur de Tab[j] au
variable max
                        posmax = j 'affectation pour prendre la position du max

                End If
                moy = moy + Tab(j) 'la calcule de la moyenne des entiers
        End If

Next

        Console.WriteLine("le maximum des entiers est : " & max & " sa
position est : " & posmax + 1) 'Affichage du resultat
        Console.WriteLine("le minimum des entiers est : " & min & " sa
position est : " & posmin + 1) 'Affichage du resultat
        Console.WriteLine("la moyenne des entiers est : " & moy / (i - 2))
'Affichage du resultat

End Sub
```

Exercice 19

' le programme suivante permet de saisir une liste de classe comprenet le nom de chaque stagiere

' et sa moyenne generale, ainsi il permet d'afficher les noms et les moyennes de tous les stagieres

'qui on une moyenne superrieur a la moyenne de la classe , plus que ça le programme doit afficher

Programmation structurée

```
'
'le nom et la moyenne du premier et du dernier de la classe
' donc on remplit un tableau a partir du saisi de l'utilisateur ;
' on fait la clacule de la moyenne general de toute les notes saisie
et on affiche
' on cherche les notes qui on plus grand et plus petit que la moyenne
, c'est pour ca on tester si
' une note est superrieur ou inferieur a la moyenne on affiche avec le
nom qui correspondant.

Sub Main()

    Dim Tnom(5) As String ' declaration d un tableau de type chaine de
caractere
    Dim Tmoy(5) As Single 'declaration d'un tableau de type réel
    Int(i) 'declaration d un variable de type entier
    Dim GrandM, PetitM, S As Single 'declaration des variables de type
réel
    Dim diligent, paresseux As String 'declaration d un variable de
type chaine de caractare

    S = 0 ' intialisation par 0

    For i = 0 To 4

        Console.WriteLine() ' //-----
        Console.WriteLine("n:" & i + 1) ' // Affich
age //
        Console.WriteLine("Nom :") ' //
de saisit //
        Tnom(i) = Console.ReadLine ' //
pour remplir //
        Console.WriteLine("Sa moyenne: ") ' //
le tableau //
        Tmoy(i) = Console.ReadLine() ' //-----
        Console.WriteLine() //

        S = S + Tmoy(i) ' calculer la somme des entiers

    Next
    S = S / 5 'calculer la moyenne

    Console.WriteLine("Qui on eu une note superrieur a la moyenne
general : ")

    for i=0 to 4

        If (S < Tmoy(i)) Then 'tester si la note est superrieur a la
moyenne de la classe

            Console.WriteLine(" " & Tnom(i) & " : " & Tmoy(i))
```

```
End If

Next

GrandM = Tmoy(0) 'initialisation du variable par la 1er moyenne
PetitM = Tmoy(0) 'initialisation du variable par la 1er moyenne
diligent = Tnom(0) 'initialisation du variable par le nom la 1er
moyenne
paresseux = Tnom(0) ' initialisation du variable par le nom la 1er
moyenne

For i = 0 To 5
    ' pour trouver le premier de la classe
    If (GrandM < Tmoy(i)) Then

        GrandM = Tmoy(i)
        diligent = Tnom(i)

    Else ' pour trouver le dernier de la classe
        If (PetitM > Tmoy(i)) Then

            PetitM = Tmoy(i)
            paresseux = Tnom(i)
        End If
    End If
Next

'      le resultat      */
Console.WriteLine("\t *****")
Console.WriteLine("# Le 1ere de la classe est :" << diligent << "
,sa moyenne est : ")
Console.WriteLine("# Le dernier de la classe est :" & paresseux &
" ,sa moyenne est : " & PetitM)
Console.WriteLine(" *****")
End Sub
```

Exercice 20

'un WAGON comporte 60 place assize dont 30 place pour les non-fumeures ,
numeroté de 1 a 30,

'et 30 place pour les fumeures numeroté de 31 à 60 .

'le programe suivant permet de faire le réservation des places des
WAGON et d'arreter s'il n y a aucune place libre

'aussi , on voudrer avoir la posibiliter d'annuler une reservation
faite par un cliet (le programme permet d'effectuer

'cette tâche sachant que le client peut oublier ou égarer le numero de
sa place .

'dans tout les cas on facilite la tâche d'annulation de la reservation
, et permettre de reserver les places annuler pour d'autre personne .

Programmation structurée

```
Sub Main()
    Dim place, i, choix, choix1 As Integer 'declaration des variables
de type entier
    Dim rep, nom, siti As String 'declaration des variables de type
chaîne de caractère
    Dim Tnum(6) As Integer 'declaration d'un tableau de type entier
avec une taille de 6
    Dim Tnom(6) As String 'declaration d'un tableau de type chaîne de
caractère avec une taille de 6

    choix = 0 'initialisation par 0
    choix1 = 0 'initialisation par 0
    i = 0 'initialisation par 0

    Do

        Console.WriteLine("    Menu du Wagon ") '
/* _____ */
        Console.WriteLine("Pour Reserver une place .Appuyer sur 1") '
/*    L E                               */
        Console.WriteLine("Pour Supprimer une place.Appuyer sur 2") '
/*          M E N U                       */
        Console.WriteLine("Pour Quitter .Appuyer sur 3") '
/* _____ */
        choix = Console.ReadLine()

        Select (choix)

            Case 1

                Console.WriteLine("Voulez-vous entrer 'OUI' si le
voyageure est fume ") 'demande a l'utilisateur de saisir si le voyageur fume
                siti = Console.ReadLine() ' lire la reponse

                If (siti = "oui") Then 'si la situation est oui

                    Do

                        If (i < 3) Then

                            If (Tnom(i) = " " And Tnum(i) = 0) Then

                                Console.WriteLine("Veuillez saisir un
nom ") 'demande a l'utilisateur de saisir le nom du voyageur
                                Tnom(i) = Console.ReadLine() 'lire le
nom
                                Console.WriteLine("Veuillez saisir le
num de la place ") 'demande a l'utilisateur de saisir le numero de la place
                                Tnum(i) = Console.ReadLine 'lire le
numero du place

                            Else

                                Console.WriteLine("Veuillez saisir un
nom ") 'demande a l'utilisateur de saisir le nom
                                Tnom(i) = Console.ReadLine 'lire le nom
```

Programmation structurée

```
        Console.WriteLine("Veuillez saisir le
num de la place ") 'demande a l'utilisateur de saisir le numero
        Tnum(i) = Console.ReadLine() ' lire le
numero

        End If
    End If

    i = i + 1 ' Incrementation du compteur

    Console.WriteLine("voulez vous faire une autre
reservation ") 'demande a l'utilisateur de saisir si il veut faire une
autre reservation

    rep = Console.ReadLine() ' la reponse
    Loop While (i < 3 And rep = "oui")
End If

If (i >= 3) Then

    Console.WriteLine(" Y a aucune place vide") ' si il
depassé le numero de la place 3

Else

    i = 3
    Do

        If (i >= 3) Then

            If (Tnom(i) = " " And Tnum(i) = 0) Then

                Console.WriteLine("Veuillez saisir un
nom ") 'demande a l'utilisateur de saisir le nom du voyageur
                Tnom(i) = Console.ReadLine() 'lire le
nom

                Console.WriteLine("Veuillez saisir le
num de la place ") 'demande a l'utilisateur de saisir le numero de la place
numero du place
                Tnum(i) = Console.ReadLine() 'lire le

            Else

                Console.WriteLine("Veuillez saisir un
nom ") 'demande a l'utilisateur de saisir le nom du voyageur
                Tnom(i) = Console.ReadLine() 'lire le
nom

                Console.WriteLine("Veuillez saisir le
num de la place ") 'demande a l'utilisateur de saisir le numero de la place
numero du place
                Tnum(i) = Console.ReadLine() 'lire le

            End If
        End If
        i = i + 1
        Console.WriteLine("voulez vous faire une autre
reservation ") 'demande a l'utilisateur de saisir si il veut faire une
autre reservation
```

Programmation structurée

```
        rep = Console.ReadLine() 'lire la reponse
    Loop While (i < 6 And rep = "oui")
End If

If (i >= 6) Then

    Console.WriteLine(" Y a aucune place vide") ' si il
depassé le numero de la place 6
End If
```

Case 2

```
Console.WriteLine("Voulez-vous supprimez a partire du
nom.....(3)" ' /*      M E N U      */
numero ...(4)" ' /* d e l a   s u p r e s s i o n   */
choix1 = Console.ReadLine()
Select Case (choix1)
```

Case 3

```
        Console.WriteLine("Veuillez saisir le nom :")
'demander a lutilisateur de saisir le nom qu il veut supprimer
nom = Console.ReadLine() ' lire la reponse

For i = 0 To 5

    If (nom = Tnom(i)) Then

        Console.WriteLine(" le nom est :" &
Tnom(i)) 'afficher le nom
le contenu par un espace
        Console.WriteLine(" Le num de place est
:" & Tnum(i)) ' afficher le numero de place
Tnum(i) = 0 'ecrasé le contenu par un
zero
```

```
        End If
    Next
```

Case 4

```
        If (choix1 = 4) Then

            Console.WriteLine("Veuillez saisir le
numero :)") 'demander a lutilisateur de saisir le numero de place qu il
veut supprimer

            place = Console.ReadLine() ' lire la
reponse

            For i = 0 To 5

                If (place = Tnum(i)) Then
```

```
Tnom(i)) 'afficher le nom
un espace
est :" & Tnum(i)) ' afficher le numero de place
un zero

                                Console.WriteLine(" le nom est :" &
                                Tnom(i) = " " 'ecrasé le contenu par
                                Console.WriteLine(" Le num de place
                                Tnum(i) = 0 'ecrasé le contenu par

                                End If
                                Next

                                End If

                                Case 3

                                Exit Select ' pour exite le menu

                                End Select

                                End Select
                                Console.WriteLine("voulez vous recommencez ?(oui/non)")
                                'demander a lutilisateur s'il veut continuer
                                rep = Console.ReadLine() 'lire la reponse

                                Loop While (rep = "oui")
                                End Sub
```

Exercice 21

```
' declaration
Dim i, j, k, min, pos As Integer
Dim tab(10) As Integer

'remplire le tabelau
For i = 0 To 9

    Console.WriteLine("veuillez saisir un entier ")
    tab(i) = Console.ReadLine
Next

For i = 0 To 9

    pos = i
    min = tab(i)

    For j = i + 1 To 10

        If (tab(j) < min) Then

            min = tab(j)
            pos = j
        End If
    Next

    For k = pos To i + 1 Step -1

        tab(k) = tab(k - 1)
    Next
```

```
        tab(i) = min
    Next

    ' affichage du tableau
    For i = 0 To 9

        Console.WriteLine("tab[" & i & "] = " & tab(i))
    Next
```

Exercice 22

```
Dim j, i, min As Integer ' declaration
Dim sortir As Boolean ' declaration
Dim tb(10) As Single ' declaration

For i = 0 To 9 'l'utilisation de la boucle for permet de remplir
le tableau tb[]

    Console.Write("Element " & i + 1 & " : ")
    tb(i) = Console.ReadLine
Next

j = 9 'initialisation du variable j par une valeur fixe

Do

    sortir = True
    For i = 0 To j

        If tb(i) > tb(i + 1) Then

            min = tb(i + 1)
            tb(i + 1) = tb(i)
            tb(i) = min
            sortir = False

        End If

    Next
    j = j - 1

Loop While (j <> 0 And sortir = False)

Console.WriteLine("le tableau trier est : ")

' Afficher le resultat du tri

For i = 0 To 9

    Console.Write(tb(i) & ",")

Next
```

Exercice 23

' Le tri par permutation est le tri du jeu de cartes.

'On parcourt le tableau jusqu'à ce que l'on trouve un élément plus petit que le précédent,

Programmation structurée

'donc mal placé. On prend cet élément et on le range à sa place dans le tableau puis on continue la lecture.
'On s'arrête à la fin du tableau.

'Soit un ensemble fini, de n éléments. Quitte à effectuer une numérotation,

'permuter les éléments de cette ensemble revient à permuter les entiers de 1 à n.

'La notation traditionnelle des permutations place les éléments qui vont être permutés dans l'ordre naturel

'sur une première ligne, et les images en correspondance, sur une deuxième ligne. Par exemple

```
Dim tp(10) As Integer
Dim min, i As Integer

' le remplissage du tableau
Console.WriteLine(" veuillez remplir le tableau avec 10 elements ")

For i = 0 To 9
    Console.Write("Element " & i + 1 & " : ")
    tp(i) = Console.ReadLine
Next

' afficher le tableau comme il est
Console.Clear()
Console.WriteLine(" le tableau donne :")

For i = 0 To 9
    Console.Write(tp(i) & "|")
Next

Console.WriteLine()

' la partie qui fait le tri par permutation */

For i = 0 To 8

    If (tp(i) > tp(i + 1)) Then

        min = tp(i + 1)
        Dim k As Integer = 0
        While (min >= tp(k))

            k = k + 1
        End While

        For j = i + 1 To k + 1 Step -1

            tp(j) = tp(j - 1)
```

```
        Next
        tp(k) = min
    End If

Next

' affichage du tableau triee */
Console.WriteLine("le tableau triee :")
For i = 0 To 9

    Console.Write(tp(i) & " | ")
Next
```

```
Console.ReadLine()
```

Exercice 24

```
Dim tcomp(10) As Integer
Dim tri As Boolean
Dim tposition(10) As Integer

' le remplissage du tableau
For i = 0 To 9

    Console.Write("Element " & i + 1 & " : ")
    tcomp(i) = Console.ReadLine
Next

Console.Clear()
Console.WriteLine("le tableau donne ")
For i = 0 To 9

    Console.WriteLine(tcomp(i) & " / ")
Next

'donner l numéro de position de ce num c'est-a-dire le nombre de
num plu min.
For i = 0 To 9

    For j = 0 To 9
        If tcomp(i) > tcomp(j) Then
            tposition(i) = tposition(i + 1)
        End If
    Next

Next

Do

    For k = 0 To 8

        Dim num As Integer = tcomp(tposition(k))
        tcomp(tposition(k)) = tcomp(k)
        tcomp(k) = num

        Dim pos As Integer = tposition(tposition(k))
```

```
tposition(tposition(k)) = tposition(k)
tposition(k) = pos

Next
tri = False
For f = 0 To 8

    If (tcomp(f) > tcomp(f + 1)) Then
        tri = True
    End If
Next
Loop While (tri = True)

' Afficher le tableau trier
Console.WriteLine("le tableau trier :")
For k = 0 To 9

    Console.WriteLine(tcomp(k) & " / ")
Next
```

Exercice 25

```
Dim i, j, m, n As Integer

For i = 14 To 0 Step -1

    For j = 0 To i

        Console.Write("*")
    Next
    Console.WriteLine()

Next

Console.ReadLine() cice 25
```

Exercice 26

```
Dim i, k, m As Integer

m = 15

For k = 0 To 14

    Console.Write(Space(m))
    For i = k To 0 Step -1

        Console.Write("*")
    Next

    m = m - 1
    Console.WriteLine()

Next
```

```
Console.ReadLine()
```

Exercice 27

```
Dim i, j As Integer
For i = 0 To 14

    For j = 0 To i

        Console.Write("*")
    Next
    Console.WriteLine()

Next
Console.ReadLine()
```

Exercice 28

```
Dim i, j, m As Integer

m = 7

For i = 0 To 13 Step 2

    Console.Write(Space(m))
    For j = 0 To i

        Console.Write("*")
    Next
    Console.WriteLine()

    m = m - 1
Next
Console.ReadLine()
```

Exercice 29

```
Dim i, j, m As Integer

m = 0

For i = 11 To 0 Step -2
    m = m + 1
    Console.Write(Space(m))
    For j = 0 To i

        Console.Write("*")
    Next
    Console.WriteLine()

Next
Console.ReadLine()
```

Exercice 30

```
Dim i, j, m, n As Integer

m = 7
n = 14
For i = 0 To n Step 2

    Console.Write(Space(m + 1))
    For j = 0 To i

        Console.Write("*")
```

```
Next
    Console.WriteLine()

    m = m - 1
Next

If ((n Mod 2) = 1) Then

    n = n + 1

End If
For k = n To 0 Step -2

    Console.Write(Space(m + 1))
    n = -1
    For i = n To k

        Console.Write("*")

    Next
    m = m + 1
    Console.WriteLine()

Next
Console.ReadLine()
```

Exercice 31

```
'Ce programme permet de decompose une somme saisie au Clavier en billets
'et pièces et affiche la decomposition.
'Par exemple :
' une somme d'argent saisie :38 Dhs sera decomposée comme suit : 1
billets de 20Dhs,10Dhs,5dhs,2Dhs,1Dhs
Sub Main()

    ' declaration des variable
    Dim rep As String
    Dim somme As Integer

    Do
        Console.Clear() 'vider l ecran

        Console.WriteLine("veuillez SVP entrez une somme d'argent")
        somme = Console.ReadLine()

        Console.Clear()
        Console.WriteLine("la decomposante du " & somme & " Dhs est :")

        Do
            Dim nbr As Integer = 0
            While (somme >= 200)

                somme = somme - 200
                nbr = nbr + 1

            End While
            If (nbr <> 0) Then
                Console.WriteLine(nbr & " billets de 200 Dhs")
            End If
        End Do
    End Do
End Sub
```

Programmation structurée

```
If (somme >= 100) Then
    somme = somme - 100
    Console.WriteLine("1 billet de 100 Dhs")
End If

End If

If (somme >= 50) Then
    somme = somme - 50
    Console.WriteLine("1 biellet de 50 Dhs")
End If

nbr = 0
While (somme > 20)
    somme = somme - 20
    nbr = nbr + 1
End While
If (nbr <> 0) Then
    Console.WriteLine(nbr & " billets de 20 Dhs")

    If (somme >= 10) Then
        somme = somme - 10
        Console.WriteLine("1 piece de 10 Dhs")
    End If
End If

If (somme >= 5) Then
    somme = somme - 5
    Console.WriteLine("1 piece de 5 Dhs")
End If

nbr = 0
While (somme >= 2)
    somme = somme - 2
    nbr = nbr + 1
End While
If (nbr <> 0) Then
    Console.WriteLine(nbr & " pieces de 2 Dhs")

    If (somme >= 1) Then
        somme = somme - 1
        Console.WriteLine("1 piece de 1 Dhs")
    End If
End If
```

```
        End If
    End If

    Loop While (somme <> 0) ' tant que la somme est differente a 0
on reboucle jusqu'a la somme est = a 0

    Console.WriteLine("Voulez-vous continuer ? (oui/non)")
    rep = Console.ReadLine()
    Loop While (rep = "oui")

End Sub
```

Exercice 32

```
Dim ent, k, i As Integer
Dim tab(10) As Integer
Dim trouve As Integer
trouve = False

For i = 0 To 9

    Console.WriteLine("Element " & i + 1 & " : ")
    tab(i) = Console.ReadLine()
Next

Console.Clear()
Console.WriteLine("veuillez saisir un entier")
ent = Console.ReadLine()

k = 0
For i = 0 To 9

    If (tab(i) = ent) Then

        trouve = True
        k = k + 1
    End If
Next
If (trouve = False) Then

    Console.WriteLine(" L'occurence n'existe pas dans le tableau ")

Else
    Console.WriteLine("L/les occurence(s) est existe : " & k & "
fois")
End If
Console.ReadLine()
```

Exercice 33

```
Dim i, j, t(6) ' declaration des variables de type entier et aussi un
tableau
Dim tab_carree(6) As Integer 'declaration d un tableau de type
entier pour ranger les elemnts qui possedent leur carree
```

```
Dim trouve As Boolean ' declaration d un variable de type boolean
trouve = False ' inisialisation par faux

For i = 0 To 6 ' le remplissage du tableau

    Console.WriteLine("Element " & i + 1 & " : ")
    t(i) = Console.ReadLine()
Next
Console.Clear()
Console.WriteLine("les elements qui possedent leur carres :")

Dim k As Integer = 0 ' declaration et inisialisation en meme temps
d un variable

For i = 0 To 6

    j = 0 ' initialisation d un variable par un 0

    Do

        If (t(i) * t(i) = t(i)) Then ' tester si le carree du
elemnts existe dans le tableau

            tab_carree(k) = t(i) 'affecter l element a un case du
tableau pour ranger les elements qui possedent leur carree
            trouve = True 'changer la valeur du trouve
            k = k + 1 'incrementation du compteur
        End If
        j = j + 1 ' incrementation du compteur
    Loop While (j < 6 And t(i) * t(i) <> t(j - 1)) ' pour sortire
de la boucle
    Next

    If (trouve = False) Then ' cette condition permet de confirmer qu
il n y a aucun elements a leur carree dans le tableau

        Console.Clear() 'vider l ecran
        Console.WriteLine("Aucun element du tableau qui possedent leur
corree ") ' l affichege s il n existe pas

    Else

        For j = 0 To k - 1 ' parcourir le tableau

            Console.WriteLine(tab_carree(j)) 'afficher les elements qui
on leur carree

            If (j < k - 1) Then 'pour ne pas afficher une ',' a la fin
                Console.WriteLine(" ,")
            End If

        Next
    End If
```

Exercice 34

```
Dim n, x, S1 As Integer
S1 = 0
Console.WriteLine("veuillez saisir un entier SVP ")
n = Console.ReadLine
```

```
Do

    x = n Mod 2
    If (x = 0) Then

        n = n / 2

    Else

        n = (n * 3) + 1
    End If
    S1 = S1 + 1
    Console.WriteLine(n)
    If (n > 1) Then
        Console.WriteLine(" ")
    End If
loop while(n>1)

Console.WriteLine("Le nombre de fois est : " & S1)
```

Exercice 35

```
Dim L, C, i, j As Integer
L = 3
C = 2
Dim M(L, C) As Integer
Dim V(L * C) As Integer

For i = 1 To L

    For j = 1 To C

        console.WriteLine("element (" & i & ", " & j & ") :")
        M(i, j) = Console.ReadLine
    Next
Next
Dim k As Integer = 0
For i = 1 To L

    For j = 1 To C

        V(k) = M(i, j)
        k = k + 1
    Next

    Next
    Console.Clear()
    Console.WriteLine("voici le nouveau tableau")

    For i = 0 To k - 1

        Console.WriteLine(V(i))

    Next
```

Exercice 36

```
Dim Annee, salaire, age, Rsalaire As Integer

Console.Write("Années d'enciente ")
```

```
Annee = Console.ReadLine
Console.Write("Votre salaire ")
salaire = Console.ReadLine
Console.Write("Votre age ")
age = Console.ReadLine

If (Annee <= 10) Then
    Rsalaire = (salaire / 2) * Annee
Else
    Rsalaire = (salaire / 2) * 10 + ((Annee - 10) * salaire)
End If
If (age > 45 And age < 49) Then

    salaire = Rsalaire + (salaire * 2)

Else

    If (age > 50) Then
        salaire = Rsalaire + (salaire * 5)
    End If

    Console.WriteLine("votre indemnite est :" & Rsalaire & " Dhs")
End If
Console.ReadLine()
```

Exercice36

```
Dim i, j As Integer

Dim A(3, 3) As Integer

For i = 1 To 3
    For j = 1 To 3
        Console.Write("element (" & i & "," & j & ") :")
        A(i, j) = Console.ReadLine
    Next
Next
Dim tA(3, 3) As Integer
For i = 1 To 3
    For j = 1 To 3
        tA(i, j) = A(i, j)
    Next
Next
Console.Clear()
Console.WriteLine("voici le nouveau tableau")

For i = 1 To 3
    For j = 1 To 3
        Console.WriteLine(tA(i, j) & " ")
    Next
    Console.WriteLine()
Next
```

Exercice 37

```
Dim P(3, 3) As Integer
Dim i, j As Integer

For i = 1 To 3

    For j = 1 To 3

        Console.WriteLine("element (" & i & "," & j & ") :")
        P(i, j) = Console.ReadLine()
    Next
Next

Dim somme As Integer = 0
Console.Clear()
Console.WriteLine("les elements sur la diagonale son :")
For i = 1 To 3

    Dim m As Integer = 0

    For j = 1 To 3

        If (i = j) Then

            Console.WriteLine(Space(m) & P(i, i)) 'pour terminer
            somme = somme + P(i, i)
        End If
        m = m + 3
    Next

    Next

    Console.WriteLine("la somme des elements sur la diagonale son : " &
    somme)

    Console.ReadLine()
```

Exercice 38

```
Dim i, j, ent As Integer

Dim tab(11) As Integer
Dim trouve As Boolean
trouve = False

For i = 0 To 9

    Console.WriteLine("element " & i + 1 & " : ")
    tab(i) = Console.ReadLine
Next

Console.WriteLine("veuillez saisir l'entier a inserer")
ent = Console.ReadLine()

Do

    If (ent <= tab(i)) Then
```

```
        trouve = True

        For j = 10 To i Step -1

            tab(j) = tab(j - 1)
        Next
        tab(i) = ent
    End If

    i = i + 1
Loop While (trouve = True And i <= 10)

For i = 0 To 10
    Console.WriteLine(tab(i))
Next
```

Exercice 39

```
Dim x, i, j As Integer
Dim A(10) As Integer

For i = 0 To 9
    Console.WriteLine("element " & i + 1 & " : ")
    A(i) = Console.ReadLine
Next

Console.WriteLine("veuillez saisir un entier ")
x = Console.ReadLine()

Dim pos As Integer = 0

For i = 0 To 9

    If (x = A(i)) Then

        For j = i To 9

            A(j) = A(j + 1)
        Next
        pos = pos + 1
    End If
Next

If (pos = 0) Then

    Console.WriteLine("rien a trouver ")
Else

    For i = pos To 9

        A(i) = 0
    Next
End If

For i = 0 To 9
    Console.WriteLine(A(i))
```

Next

Exercice 40

```
Dim s As Integer = 12
    Dim X As Integer = 1
    Dim n As Integer = 0
    Dim i As Integer = 0
    Dim k As Integer = 0
    Dim m As Integer = 20
    Dim l As Integer = 5
    Dim multi As Integer = 0

    For i = 0 To 7

        If (i <= 0) Then

            Console.WriteLine(i + 1 & " * 8 =" & 8 + 1)
        End If

        n = 11 * (X * 10) + k + 1
        multi = s
        multi = multi * 8 + i + 2
        Console.WriteLine(s & " * 8 =" & multi)
        s = s + n
        k = 10
        X = X * 10

    
```

Next

```
Console.ReadLine()
```

Exercice 41

```
Dim i, j As Integer ' declaration des variables de type entier
Dim tab(50) As Char ' declaration d un tableau char et initialiser
par des espaces

Console.WriteLine("Veuillez saisir une phrase qui termine par un
'.' ")
For i = 0 To 50
    tab(i) = Console.ReadLine()
Next

j = 0 ' initialisation par 0
i = 0 ' initialisation par 0

Do

    Console.Write("Mot " & j + 1 & " : ") ' au moins on affiche un
mot
    Do

        ' si la valeur de la case du tableau est different a
espace on affiche cette valeur , et aussi different un '.' ( terminateur)
        If (tab(i) <> " " And tab(i) <> ".") Then
```

```
        tab(i) = Console.ReadLine 'affichage des caracteres
    End If
    i = i + 1 'incrementation du compteur
    Loop While (tab(i - 1) <> " " And tab(i - 1) <> ".") ' la
condition dans while pour sortire de la boucle

    j = j + 1 ' incrementation du compteur
    Console.WriteLine() ' apres chaque mot , on est besoin de
retourner a la ligne
    Loop While (i <> 50 And tab(i - 1) <> ".") ' la condition dans
while pour sortire de la boucle
    Console.ReadLine()
```

Exercice 42

```
Function F(ByRef t1() As Integer, ByRef t2() As Integer, ByVal n As Integer)
As Integer
```

```
    Dim nb, i, j As Integer
```

```
    nb = 0
    j = 0
    Console.WriteLine("les valeurs sont : ")
    For i = 0 To n
```

```
        If (t1(i) > 0 And t1(i) < 10) Then
```

```
            nb = nb + 1
            Console.WriteLine(t1(i) & " / ")
            t2(j) = t1(i)
            j = j + 1
```

```
        End If
```

```
    Next
```

```
    Return nb
```

```
End Function
```

```
Sub Main()
```

```
    Dim n As Integer = 10
    Dim tab(n), i, X As Integer
```

```
    Dim t2(n) As Integer
    For i = 0 To 9
```

```
        Console.WriteLine("Element " & i + 1 & " : ")
        tab(i) = Console.ReadLine
```

```
    Next
```

```
    Console.Clear()
```

```
    Console.WriteLine(" le tableau d'origine est : ")
```

```
    For i = 0 To 9
```

```
        Console.WriteLine(tab(i) & " | ")
```

```
    Next
```

```
    X = F(tab, t2, n)
```

```
    Console.WriteLine("le nombre de valeurs comprises entre 0 et 10
dans le tableau : " & X)
```

Exercice 43

```
Dim i, j As Integer

    Dim Tableau(3, 4)
    For i = 0 To 3

        For j = 0 To 4

            Console.WriteLine("element (" & i & ", " & j & ") :")
            Tableau(i, j) = Console.ReadLine

        Next

    Next

    Dim min As Integer = Tableau(0, 0)
    Dim max As Integer = Tableau(0, 0)

    For i = 0 To 3

        For j = 0 To 4

            If (Tableau(i, j) < min) Then

                min = Tableau(i, j)

            Else

                If (Tableau(i, j) > max) Then

                    max = Tableau(i, j)

                End If

            End If

        Next

    Next

    Console.WriteLine("le minimum des entiers est : " & min)
    Console.WriteLine("le maximum des entiers est : " & max)
```

Exercice 44

```
Dim i, j As Integer
' la boucle de l affichage horizontale*/
Console.Write("X*Y  ")
For j = 0 To 10
    Console.Write(" " & j)
Next
Console.WriteLine()
Console.WriteLine("-----")
-----")
'la boucle de l affichage vertical*/
For j = 0 To 10
' decalage pour le dernier ligne*/
    If (j <> 10) Then
```

```
        Console.WriteLine(" " & j)
    Else
        Console.WriteLine(j)
    End If

    ' la multiplication */
    For i = 0 To 10 'boucle pour
        Console.WriteLine(j * i & " ") 'affichage de la tabelle de
multiplication
    Next

    Console.WriteLine()
Next

Console.ReadLine()
```

Exercice 45

```
Dim tab(3) As Integer
Dim tab2(3) As Integer
Dim i, multi As Integer

multi = 0

For i = 0 To 2

    Console.WriteLine("element " & i + 1 & " : ")
    tab(i) = Console.ReadLine

Next
For i = 0 To 2

    Console.WriteLine("element " & i + 1 & " : ")
    tab2(i) = Console.ReadLine
Next
Console.Clear()
Console.WriteLine("( ")
For i = 0 To 2
    Console.WriteLine(tab(i) & " ")
Next
Console.WriteLine(") * ( ")

For i = 0 To 2
    Console.WriteLine(tab2(i) & " ")
Next
Console.WriteLine(" ) =")

For i = 0 To 2

    multi = tab(i) * tab2(i)
    If (multi > 0) Then

        Console.WriteLine(multi & "+")

    Else

        Console.WriteLine(multi)
```

```
End If

Next
multi = multi + multi
Console.WriteLine(" = " & multi)
Console.ReadLine()
```

Exercice 46

```
' Déclarations des tableaux et leurs dimensions */
Dim A(50), B(50), FUS(100) As Integer
Dim N, M, iA, iB, iFUS As Integer

' Saisie des données du tableau A */
Console.WriteLine("Dimension du tableau A : ")
N = Console.ReadLine
Console.WriteLine("Entrer les éléments de A dans l'ordre
croissant :")
For iA = 0 To N - 1

    Console.WriteLine("Élément A[" & iA + 1 & "] : ")
    A(iA) = Console.ReadLine
Next

' Saisie des données du tableau B*/
Console.WriteLine("Dimension du tableau B")
M = Console.ReadLine

Console.WriteLine("Entrer les éléments de B dans l'ordre
croissant :")
For iB = 0 To M - 1

    Console.WriteLine("Élément B[" & iB + 1 & "] : ")
    B(iB) = Console.ReadLine
Next

' Fusion des éléments de A et B dans FUS */

iA = 0
iB = 0
iFUS = 0

While (iA < N And iB < M)
    If (A(iA) < B(iB)) Then

        FUS(iFUS) = A(iA)
        iFUS = iFUS + 1
        iA = iA + 1

    Else

        FUS(iFUS) = B(iB)
        iFUS = iFUS + 1
        iB = iB + 1
    End If
End While
'pour copier le reste du tableau A si il n'est pas encor a la fin
*/
While (iA < N)
```

```
        FUS(iFUS) = A(iA)
        iFUS = iFUS + 1
        iA = iA + 1
    End While
    'pour copier le reste du tableau B si il n'est pas encor a la fin
*/
    While (iB < M)

        FUS(iFUS) = B(iB)
        iFUS = iFUS + 1
        iB = iB + 1
    End While

    Console.WriteLine("Tableau FUS trieé par ordre croissant :")
    For iFUS = 0 To (N + M) - 1

        Console.WriteLine(FUS(iFUS))

        If (iFUS < (N + M) - 1) Then
            Console.Write("/")
        End If
    Next
```

Exercice 47

```
' Déclarations */
    Dim P(20, 20) As Integer
    Dim i, j, N As Integer

    'Saisie des données */
    Do
        Console.WriteLine("veuillez entrez le degrer N du triangle
( entre 1 et 19) : ")
        N = Console.ReadLine
    Loop While (N > 19 Or N < 0)

    For i = 0 To N

        P(i, 0) = 1 'pour commencé avc 1
        For j = 1 To i

            P(i, j) = P(i - 1, j) + P(i - 1, j - 1) 'pour faire l
addition
        Next

        P(i, i) = 1 'pour terminer avc 1
    Next
    ' résultat */

    Console.WriteLine("Triangle de Pascal de degrer " & N)

    For i = 0 To N
```



```
Next
Console.Write("|")
Console.WriteLine()
Console.Write(" \\  
/")
```

Exercice 49

```
Dim i, j As Integer
```

```
Dim M(3, 4) As Char = {"a", "b", "c", "d", "e", "f", "g", "h", "i",  
"j", "k", "l"}  
Dim V(4, 3) As Char
```

```
Dim K As Integer = 0  
For i = 0 To 3  
  
    For j = 0 To 4  
  
        V(i, j) = Mid(i, j)
```

```
Next
```

```
Console.WriteLine()  
Next  
Console.Clear()
```

```
Console.Write(" / \\  
")  
Console.WriteLine()
```

```
For i = 0 To 3  
    Console.Write("|")  
    For j = 0 To 3  
  
        Console.Write(Mid(i, j) & " ")
```

```
Next
```

```
Console.Write("|")  
Console.WriteLine()
```

```
Next  
Console.Write(" \\  
/")
```

```
'////////////////////////////////////
```

```
Console.WriteLine(" / \\  
")  
Console.WriteLine()
```

```
For i = 0 To 4  
    Console.Write("|")  
    For j = 0 To 3
```

```
        Console.WriteLine(V(j, i) & " ")
    Next
    Console.Write("|")
    Console.WriteLine()

Next

Console.Write(" \\      /")
```

Exercice 50

```
Dim i, j As Integer

Dim M(3, 4) As Char = {"a", "b", "c", "d", "e", "f", "g", "h", "i",
"j", "k", "l"}
Dim V(4, 3) As Char

Dim K As Integer = 0
For i = 0 To 3

    For j = 0 To 4

        V(i, j) = Mid(i, j)

    Next

    Console.WriteLine()
Next
Console.Clear()

Console.Write(" /      \\")
Console.WriteLine()

For i = 0 To 3
    Console.Write("|")
    For j = 0 To 3

        Console.Write(Mid(i, j) & " ")

    Next

    Console.Write("|")
    Console.WriteLine()
Next
Console.Write(" \\      /")

'////////////////////////////////////

Console.WriteLine(" /      \\")
Console.WriteLine()

For i = 0 To 4
    Console.Write("|")
    For j = 0 To 3
```

```
        Console.WriteLine(V(j, i) & " ")
    Next
    Console.Write("|")
    Console.WriteLine()

Next

Console.Write(" \\    /")
```

Exercice 51

```
Dim tab(3) As Integer
Dim tabl(3) As Integer
Dim multil, multi, i As Integer

For i = 0 To 2

    Console.WriteLine("Element " & i + 1 & " : ")
    tab(i) = Console.ReadLine

Next

For i = 0 To 2

    Console.WriteLine("Element " & i + 1 & " : ")
    tabl(i) = Console.ReadLine

Next

multi = 0

For i = 0 To 2
    For j = 0 To 2
        multi = tab(i) * tabl(j)
        Console.WriteLine(multi)
        multil = multi + multi
    Next
Next

Console.WriteLine("l'addition de tout les multiplication des
schtroupmf : " & multil)
Console.ReadLine()
```

Exercice 52

```
Dim tab(10), i As Integer

Console.WriteLine("Veuillez taper 10 entier compris entre 0 et 20
")

For i = 0 To 10

    Console.WriteLine("Element[" & i + 1 & "] : ")
    tab(i) = Console.ReadLine
Next

For k = 0 To 10
    Dim nb As Integer = 0
    For i = 0 To 10

        If (tab(i) = k) Then
```

```
        nb = nb + 1

    End If

Next
Console.WriteLine(nb & " fois que vous tapez " & k)

Next
```

Exercice 53

```
' Déclarations des tableaux et leurs dimensions
Dim T(50), V(50), R(100) As Integer
Dim N, M, iT, iV, iR As Integer

' Saisie des données du tableau A
Console.WriteLine("Dimension du tableau A : ")
N = Console.ReadLine
Console.WriteLine("Entrer les éléments de A dans l'ordre
croissant :")
For iT = 0 To N - 1

    Console.WriteLine("Élément A[" & iT + 1 & "] : ")
    T(iT) = Console.ReadLine
Next

' Saisie des données du tableau B
Console.WriteLine("Dimension du tableau B")
M = Console.ReadLine

Console.WriteLine("Entrer les éléments de B dans l'ordre
croissant :")
For iV = 0 To M - 1

    Console.WriteLine("Élément B[" & iV + 1 & "] : ")
    V(iV) = Console.ReadLine
Next

' Fusion des éléments de A et B dans FUS */

iT = 0
iV = 0
iR = 0
While (iT < N And iV < M)
    If (T(iT) < V(iV)) Then

        R(iR) = T(iT)
        iR = iR + 1
        iT = iT + 1

    Else

        R(iR) = V(iV)
        iR = iR + 1
        iV = iV + 1
    End If
```

```
End While
' pour copier le reste du tableau T si il n'est pas encor a la fin
While (iT < N)

    R(iR) = T(iT)
    iR = iR + 1
    iT = iT + 1
End While
' pour copier le reste du tableau V si il n'est pas encor a la fin
While (iV < M)

    R(iR) = V(iV)
    iR = iR + 1
    iV = iV + 1
End While

Console.WriteLine("Tableau FUS trieé par ordre croissant :")
For iR = 0 To (N + M) - 1

    Console.Write(R(iR))
    If (iR < (N + M) - 1) Then
        Console.Write("/")
    End If
Next
Console.ReadLine()
```

Exercice 56

```
Dim tab1(5) As Integer
Dim tab2(5) As Integer
Dim i As Integer
Console.WriteLine(" Remplire le 1er tableau ")
For i = 0 To 4

    Console.WriteLine("elemnet" & i + 1 & " : ")
    tab1(i) = Console.ReadLine
Next

Console.WriteLine(" Remplire le 2eme tableau ")
For i = 0 To 4

    Console.WriteLine("elemnet" & i + 1 & " : ")
    tab2(i) = Console.ReadLine
Next

i = 0
Dim j As Integer
Do

    Do

        j = 0
        If tab1(i) <> tab2(j) Then

            Console.WriteLine(" le premier élément commun est : "
& tab1(i))

        End If

    End Do

Next
```

```
        j = j + 1
    Loop While (tab1(i) <> tab2(j - 1) And tab2(j - 1) > 5)
    i = i + 1
Loop While (tab1(i - 1) <> tab2(j - 1) And tab2(j - 1) > 5)
```

Exercice 57

```
Dim TAB() As Char
Char={ "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z", "a" };
Dim phrase(50) As Char

Console.WriteLine("entrer une chaine : ")
phrase = Console.ReadLine
Dim X As Integer = Len(phrase)
Console.Clear()

Console.WriteLine("la chaine : " & phrase & " , sera : ")

Dim i As Integer = -1
Dim j As Integer
Do

    i = i + 1
    For j = 0 To 26

        If (phrase(i) == TAB(j) And phrase(i) <> " ") Then

            phrase(i) == TAB(j + 1)
            Console.WriteLine(phrase(i))

        End If
    Next

    If (phrase(i) = " ") Then
        Console.WriteLine(" ")
    End If
loop while (i <> X )
```

Exercice 58

```
Dim TAB() As Char = {"a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z", "a"}
Dim phrase(50) As Char

Console.WriteLine("entrer une chaine : ")
phrase = Console.ReadLine
Dim X As Integer = Len(phrase)
Console.Clear()

Console.WriteLine("la chaine : " & phrase & " , sera : ")

Dim dec As Integer
```

```
Do
    Console.WriteLine("Entrer un entier pour le decalage (entre 1
et 13)")
    dec = Console.ReadLine
Loop While (dec < 1 Or dec > 13)

Console.Clear()
Console.WriteLine("la chaine : " & phrase & " , sera : ")

Dim i As Integer = -1
Do
    i = i + 1
    For j = 0 To 25

        If (phrase(i) = TAB(j) And phrase(i) <> " ") Then

            phrase(i) = TAB(dec + j)
            Console.Write(phrase(i))

        End If
    Next

    If (phrase(i) = " ") Then
        Console.WriteLine(" ")
    End If
Loop While (i <> X)
```

Exercice 59

```
Dim TAB() As Char = {"a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k",
"l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z",
"a"}
Dim code() As Char = {"H", "Y", "L", "U", "J", "P", "V", "R", "E",
"A", "K", "B", "N", "D", "O", "F", "S", "Q", "Z", "C", "W", "M", "G", "I",
"T", "X"}
Dim phrase(50) As Char

Console.WriteLine("entrer une chaine : ")
phrase = Console.ReadLine
Dim X As Integer = Len(phrase)
Console.Clear()

Console.WriteLine("la chaine : " & phrase & " , sera : ")

Dim j As Integer
Dim i As Integer = -1
Do
    i = i + 1
    For j = 0 To 26
```

```
    If (phrase(i) = TAB(j) And phrase(i) <> " ") Then
        phrase(i) = code(j)
        Console.Write(phrase(i))
```

```
    End If
Next
```

```
    If (phrase(i) = " ") Then
        Console.Write("/")
    End If
```

```
loop while(i<>X )
```

Exercice 60

```
Dim phrase(50) As Char
    Dim c As Char
```

```
Console.WriteLine("entrer une chaine : ")
phrase = Console.ReadLine
Dim X As Integer = Len(phrase)
Console.Clear()
```

```
Console.WriteLine("Entrer le caracter : ")
c = Console.ReadLine
```

```
Dim i As Integer = 0
Dim nbr As Integer = 0
Do
```

```
    For i = 0 To X
```

```
        If (c = phrase(i)) Then
```

```
            For j = i To X - 1
```

```
                phrase(i) = phrase(i + 1)
```

```
            Next
```

```
        End If
```

```
    Next
```

```
        i = i + 1
Loop While (i < X)
```

```
For i = 0 To X
    Console.Write(phrase(i))
```

```
Next
```

Exercice 61

```
Sub doublon(ByVal tab() As Char, ByVal t() As Char, ByRef l As Integer)
    Dim j, i, a As Integer
    l = 0
    t(0) = tab(0)
    For i = 0 To 19
        a = 0
        j = 0
        Do

            If (tab(i) = t(j)) Then

                a = a + 1
            End If
            j = j + 1

        Loop While (j < l And tab(i) <> t(j))
        If (a = 0) Then

            t(l + 1) = tab(i)
            l = l + 1
        End If
    Next
End Sub
Sub Main()
    Dim tab(20) As Char
    Dim t(20) As Char
    Dim i, l As Integer
    Console.WriteLine("saisir une phrase :")
    tab = Console.ReadLine
    doublon(tab, t, l)
    Console.WriteLine("la phrase avec supprimant les doublons :")
    For i = 0 To l

        Console.Write(t(i))
    Next

End Sub
```

Exercice 62

```
Dim t1(50) As Char
    Dim t2(50) As Char
    Dim j As Integer
    Dim i As Integer = -1
    Dim lg As Integer = 0
    Console.WriteLine("veuillez saisir un phrase ")
    t1 = Console.ReadLine
    Dim K As Integer = Len(t1)
    Console.WriteLine("veuillez saisir une deuxieme phrase ")
    t2 = Console.ReadLine
    Dim H As Integer = Len(t2)

    Do

        i = i + 1
        For j = 0 To H

            If (t1(i) = t2(j)) Then
```

```
        lg = 1
    End If
Next
Loop While (t1(i) <> t2(j) And i <> 50)

If (lg = 1) Then
    Console.WriteLine("les deux phrase sont equivalent ")
Else
    Console.WriteLine("les deux phrase ne son pas equivalent")
End If
```

Exercice 63

```
Dim tab(50) As Char
Dim mot(20) As Char

Dim trouve As Boolean
Console.WriteLine("Taper une phrase terminer avec '.'")
tab = Console.ReadLine
Dim X As Integer = Len(tab)
Console.WriteLine("Taper un mot ")
mot = Console.ReadLine
Dim Y As Integer = Len(mot)

Dim nbr As Integer = 0
Dim j As Integer = 0
Dim i As Integer = 0
Do

    Do

        trouve = False
        If (tab(i) = mot(j) And tab(i) <> " ") Then

            nbr = nbr + 1
            trouve = True
        End If
        i = i + 1
    Loop While (i - 1 = X And trouve = True)
    j = j + 1
Loop While (j <> Y)

If (nbr = X) Then
    Console.WriteLine("Le Mot :" & mot & " est eparpillement")
Else
    Console.WriteLine("Le Mot :" & mot & " est Non-eparpillement")
End If
```

Exercice 64

```
Dim phrase(80) As Char
Console.WriteLine("Entrer une chaine: ")
phrase = Console.ReadLine
Dim l As Integer = Len(phrase)

Dim pln As Boolean = True

For i = 0 To l / 2
```

```
If (pln = True) Then
    If (phrase(i) <> phrase(l - i - 1)) Then
        pln = False
    End If
End If
Next
```

```
If (pln = True) Then
    Console.WriteLine("Palindrome")
Else
    Console.WriteLine("Non Palindrome")
End If
```

Exercice 65

```
Dim N As Integer
Console.Write("Veuillez saisir une entier : ")
N = Console.ReadLine

Dim k As Single = 1
Dim i As Single
For i = 1 To N - 1
    k = (1 / i)
    k = k * k
Next
Console.WriteLine("K : " & k)
Console.ReadLine()
```

Exercice 66

```
Dim Mnt As Integer

Console.WriteLine("veuillez saisir le montant d'achat ")
Mnt = Console.ReadLine

If (Mnt > 5000) Then
    Mnt = Mnt - (Mnt * 20 / 100)
    Console.WriteLine("votre reduction est de 20%, donc : " & Mnt &
" Dhs")
Else
    If (Mnt > 3000 And Mnt <= 5000) Then
        Mnt = Mnt - (Mnt * (15 / 100))
```

```
        Console.WriteLine("votre reduction est de 15%, donc : " &
Mnt & " Dhs")
        Else

            If (Mnt > 1000 And Mnt <= 3000) Then

                Mnt = Mnt - (Mnt * 10 / 100)
                Console.WriteLine("votre reduction est de 10%, donc : "
& Mnt & " Dhs")

            Else
                Console.WriteLine("Aucune reduction pour votre montant")
            End If

        End If

    End If
    Console.ReadLine()
```

Exercice 67

```
Dim prix, cmd, Mnt As Integer
    Console.WriteLine(" Entrer le prix Unitaire : ")
    prix = Console.ReadLine
    Console.WriteLine(" Entrer la quantité de commande : ")
    cmd = Console.ReadLine

    Mnt = prix * cmd

    Console.ReadLine()
    If (Mnt > 1000) Then

        Mnt = Mnt - (Mnt * 10 / 100)
        Console.WriteLine("Le port est gratuit ")
        Console.WriteLine("la remise est de 10% ")
        Console.WriteLine("Le prix a payer : " & Mnt & " Dhs")

    Else

        If (Mnt < 1000) Then

            Console.WriteLine("Le port est de 3% ")
            If (Mnt > 300) Then

                Mnt = Mnt - (Mnt * 5 / 100)
                Console.WriteLine(" la remise est de 5%")
            End If
            Console.WriteLine("Le prix a payer : " & Mnt & " Dhs")
        End If

    End If
    Console.ReadLine()
```

Exercice 68

Exercice 69

Exercice 70

```
Dim suite(7) As Integer
```

```
Dim i As Integer
suite(0) = 1
suite(1) = 1

For i = 2 To 6

    suite(i) = suite(i - 1) + suite(i - 2)
Next

For i = 0 To 6
    Console.WriteLine(suite(i) & "/")
Next
Console.ReadLine()
```

Exercice 71

```
Dim i, j, k, pos, coef, coef1 As Integer
Dim tab(5) As Integer
Dim Tnom(5) As String ' declaration d un tableau de type chaine de
caractere
Dim Tprenom(5) As String ' declaration d un tableau de type chaine
de caractere
Dim S, Tmoy(5), min, Tnote(5) As Single 'declaration d'un tableau
de type réel

Dim note As Single = 1
Dim notel As Single = 0
For i = 0 To 4

    Console.WriteLine("//-----")
    Console.WriteLine("//")
    Console.WriteLine(" Stagiaire N:" & i + 1 & ":")
    Console.WriteLine()

    Console.WriteLine("Nom  :")
    Tnom(i) = Console.ReadLine
    Console.WriteLine("prenom  :")
    Tprenom(i) = Console.ReadLine
    coef1 = 0
    notel = 0
    For j = 0 To 4

        Console.WriteLine("Note : " & i + 1)
        Tnote(i) = Console.ReadLine
        Console.WriteLine("Coef : " & i + 1)
        coef = Console.ReadLine

        note = coef * Tnote(i)
        notel = note + notel
        coef1 = coef + coef1
    Next

    Tmoy(i) = notel / coef1 'calculer la somme des entiers

Next

For i = 0 To 5
```

```
pos = i
min = Tmoy(i)

For j = (i + 1) To 5

    If (Tmoy(j) > min) Then

        min = Tmoy(j)
        pos = j
    End If
Next

For k = pos To i + 1 Step -1

    Tmoy(k) = Tmoy(k - 1)
Next

Tmoy(i) = min
Next

'affichage du tableau */

For i = 0 To 4

    Console.WriteLine("Classement :" & i + 1)
    Console.WriteLine("Nom :")
    Tnom(i) = Console.ReadLine
    Console.WriteLine("prenom :")
    Tprenom(i) = Console.ReadLine

    For j = 0 To 4

        Console.WriteLine("Note : " & i + 1)
        Tnote(i) = Console.ReadLine
    Next
    Console.WriteLine("Moyenne : " & Tmoy(i))

Next
Console.ReadLine()
```

Exercice 72

```
Dim i, j As Integer
Dim T(10) As Integer
Dim S(10) As Integer

For i = 0 To 9

    Console.WriteLine("Element(" & i + 1 & "): ")
    T(i) = Console.ReadLine
Next
For i = 0 To 9

    S(i) = T(i)
Next
Dim nbr As Integer = 0
Do

    For i = 0 To 9
```

```
For j = 0 To 9
    If T(i) = S(j) Then
        nbr = nbr + 1
        For k = i To 9
            T(i) = T(i + 1)
        Next
    End If
Next
Next
Loop While (j < 10 And T(i) <> S(j))

Console.WriteLine("s'=[")
For i = 0 To 9
    Console.WriteLine(T(i) & ",")
Next

Console.WriteLine("]")
Console.WriteLine()
Console.WriteLine("l'=" & 10 - nbr)
Console.ReadLine()
```

Exercice 73

```
Dim tab(10), i, nbr As Integer
Dim rep As String
Dim trouve As Boolean = False

Do

    For i = 0 To 9

        Console.WriteLine("Element [" & i + 1 & "] : ")
        tab(i) = Console.ReadLine
    Next

    nbr = 0
    i = 0

    While (tab(i) < tab(i + 1) And i < 8)

        nbr = nbr + 1
        i = i + 1
        trouve = True
    End While
    If (nbr = 8) Then
        Console.WriteLine("le tableau est croissant ")
    End If

    i = 0
    nbr = 0
    trouve = False
    While (tab(i) > tab(i + 1) And i < 8)

        nbr = nbr + 1
        i = i + 1
        trouve = True
```

```
End While
If (nbr = 8) Then
    Console.WriteLine("le tableau est décroissant ")
End If
i = 0
nbr = 0
trouve = False
While (tab(i) = tab(i + 1) And i < 8)

    nbr = nbr + 1
    i = i + 1
End While
If (nbr = 8) Then
    Console.WriteLine("le tableau est constant ")
End If
If (nbr < 8 Or trouve = False) Then
    Console.WriteLine("le tableau est quelconque ")
End If

Console.WriteLine("Voulez-vous recommencez?(oui/non): ")
rep = Console.ReadLine
```

```
Loop While (rep = "oui")
```

Exercice 74

```
Dim i, j As Integer
Dim Maj() As Char = {"A", "B", "C", "D", "E", "F", "G", "H", "I",
"J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X",
"Y", "Z"}
Dim min() As Char = {"a", "b", "c", "d", "e", "f", "g", "h", "i",
"j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x",
"y", "z"}

Dim phrase(26) As Char
Console.WriteLine("Entrer une phrase :")
phrase = Console.ReadLine
Dim X As Integer = Len(phrase)

i = 0

Do

    j = 0

    Do

        If (phrase(i) = min(j)) Then

            phrase(i) = Maj(j)

            Console.WriteLine(phrase(i))

        End If

        j = j + 1
```

```
        Loop While (phrase(i) <> min(j + 1))

        i = i + 1
    Loop While (i - 1 <> X)

    Console.ReadLine()
```

Exercice 75

```
Function Vooy(ByVal voy() As Char, ByVal phrase() As Char, ByVal X As Integer) As Integer
```

```
    Dim nbr As Integer = 0
    Dim i As Integer = 0
    Dim j As Integer
    Do
        j = 0
        Do

            If (voy(i) = phrase(i)) Then

                phrase(j) = phrase(j + 1)
                nbr = nbr + 1

            End If

            j = j + 1

        Loop While (voy(i) <> phrase(j - 1))
        i = i + 1
    Loop While (voy(i - 1) <> phrase(j))

    Return nbr
End Function
```

```
Sub main()

    Dim i, j As Integer
    Dim voy() As Char = {"a", "e", "i", "u", "o", "y"}
    Dim phrase(50) As Char

    Console.WriteLine("Entrer une phrase : ")
    phrase = Console.ReadLine
    Dim X As Integer = Len(phrase)
    Dim vr As Integer = Vooy(voy, phrase, X)

    For i = 0 To X - vr
        Console.WriteLine(phrase(i))
        Console.WriteLine()
    Next
```

Exercice 76

```
Dim tab(20) As Char 'Declaration d'un tableau de type caractere

    Console.WriteLine("Tapez un verbe du premier groupe : ")
```

```
tab = Console.ReadLine
Dim i As Integer
i = Len(tab)
If (i <= 2 Or tab(i - 1) <> "r" Or tab(i - 2) <> "e") Then ' //une
condition pour tester si le verb se termine par er à la fin
    Console.WriteLine("le verbe n'est pas du premier groupe")
Else ' si il se termine par "er"

    Dim Vrb1(i - 3) As Char
    For x As Integer = 0 To i - 3

        Vrb1(i) = tab(i)

        Console.WriteLine("Le verbe est du 1ere groupe :")
        Console.WriteLine("Je      " & Vrb1 & "e")
        Console.WriteLine("Tu      " & Vrb1 & "es")
        Console.WriteLine("Il/Elle  " & Vrb1 & "e")
        Console.WriteLine("Nous     " & Vrb1 & "ons")
        Console.WriteLine("Vous    " & Vrb1 & "ez")
        Console.WriteLine("Ils/Elles " & Vrb1 & "ent")
    Next
End If
Console.readline
```

Exercice 77

```
' //////////////////////////////////////////////////// la liste des produit
////////////////////////////////////
Sub Listee(ByVal liste() As String, ByVal X As Integer)
    Dim i As Integer

    Console.Clear()
    For i = 0 To X - 1
        Console.WriteLine("( " & i + 1 & " ) " & liste(i))
    Next
End Sub
'//////////////////////////////////// Ajouter un nouveau produit
////////////////////////////////////
Function Ajou(ByVal liste() As String, ByVal code() As String, ByVal
nbr As Integer, ByVal Qnt() As Integer) As Integer

    Console.Clear()
    Dim i As Integer = 0
    Dim rep As String
    Do

        Console.WriteLine("Entrer le Code de produit : ")
        code(i) = Console.ReadLine
        Console.WriteLine("Entrer l'intitulé de produit : ")
        liste(i) = Console.ReadLine
        Do

            Console.WriteLine("Entrer la quantite : ")
            Qnt(i) = Console.ReadLine
        Loop While (Qnt(i) = 0)
        i = i + 1
    
```

```
        nbr = nbr + 1
        Console.Clear()
        Console.WriteLine("Ajouter un autre produit ? oui/non")
        rep = Console.ReadLine
    Loop While (rep = "oui")

    Return nbr

End Function
'////////// supprimer un produit partir de son code//////////
Sub suppr(ByVal liste() As String, ByVal code() As String, ByVal X As
Integer)

    Console.Clear()

    Dim cod As String
    Console.WriteLine("Code : ")
    cod = Console.ReadLine
    Dim i As Integer = -1
    Do

        i = i + 1
        If (cod = code(i)) Then

            code(i) = "0"
            Liste(i) = "0"

        End If

    Loop While (cod = code(i) And i = X)

End Sub
'//////////Achete un produit //////////
Sub Ach(ByVal liste() As String, ByVal Code() As String, ByVal prix()
As Integer, ByVal X As Integer)

    Console.Clear()
    Dim cod As String
    Dim qnt, i As Integer
    i = -1
    Console.WriteLine("Code Produit : ")
    cod = Console.ReadLine

    Do

        i = i + 1
        If cod = Code(i) Then

            Console.WriteLine("Quantitee achetee : ")
            qnt = Console.ReadLine

            Console.Clear()

            Console.WriteLine("Code : " & Code(i) & "\t\t Produit : " &
liste(i) & "\t\t Quantiter : " & qnt)
            qnt = qnt * prix(i)
            Console.WriteLine("Le prix : " & qnt)

        End If

    End If
```

Programmation structurée

```
Loop While (cod <> Code(i) And i <= X)
End Sub
'//////////////////// Vendre un produit
Sub Vnd(ByVal liste() As String, ByVal Code() As String, ByVal prix()
As Integer, ByVal X As Integer)

    Console.Clear()
    Dim cod As String
    Dim qnt, i As Integer
    i = -1
    Console.WriteLine("Code Produit : ")
    cod = Console.ReadLine

    Do

        i = i + 1
        If cod = Code(i) Then

            Console.WriteLine("Quantitee vendu : ")
            qnt = Console.ReadLine

            Console.Clear()

            Console.WriteLine("Code :" & Code(i) & " Produit : " &
liste(i) & " Quantiter : " & qnt)
            qnt = qnt * prix(i)
            Console.WriteLine("Le prix : " & qnt)

        End If
    Loop While (cod <> Code(i) And i <= X)
End Sub

Sub Main()
    Dim rep, liste(10) As String
    Dim Code(10) As String
    Dim Qnt(10) As Integer
    Dim prix() As Integer = {100, 101, 105, 50, 180, 200, 320, 150, 172,
111}
    Dim choix, X, nbr As Integer
    Do

        'Affichage du menu
        Console.WriteLine("/-/-/-/-/-/_-[MENU]-_/-/-/-/-/-/")
        Console.WriteLine(" Entrer votre choix!")
        Console.WriteLine("-1- Ajouter un produit ")
        Console.WriteLine("-2- Afficher la liste de produits")
        Console.WriteLine("-3- Supprimer un produit")
        Console.WriteLine("-4- Acheter un produit")
        Console.WriteLine("-5- Vendre un produit")
        Console.WriteLine("-6- Quitter")
        Console.WriteLine("SVP saisie votre choix : ")
        choix = Console.ReadLine
        Select Case (choix)

            Case 1
                nbr = 0
                X = Ajou(liste, Code, nbr, Qnt)

            Case 2
```

```
Listee(liste, X)

Case 3
  suppr(liste, Code, X)

Case 4
  Ach(liste, Code, prix, X)

Case 5
  Vnd(liste, Code, prix, X)

Case 6
  Exit Select

End Select
Console.WriteLine("Recommencer ?")
rep = Console.ReadLine

Loop While (rep = "oui")
End Sub
```

Exercice 78

```
Public Structure point
```

```
    Dim Aba As Single
    Dim Abb As Single
    Dim Ora As Single
    Dim Orb As Single
End Structure
Dim CO As point
Sub main()

    Console.WriteLine("Entrer les coordonnes du point A:")
    Console.WriteLine("Abscisse : ")
    CO.Aba = Console.ReadLine
    Console.WriteLine("Ordonne : ")
    CO.Ora = Console.ReadLine
    Console.WriteLine("Entrer les coordonnes du point B:")
    Console.WriteLine("Abscisse : ")
    CO.Abb = Console.ReadLine
    Console.WriteLine("Ordonne : ")
    CO.Orb = Console.ReadLine

    Dim D As Single = Sqrt((CO.Aba - CO.Abb) + (CO.Ora - CO.Orb))
    Console.Write(D & "/")

    Console.WriteLine("A(" & CO.Aba & "," & CO.Ora & ")")
    Console.WriteLine("B(" & CO.Abb & "," & CO.Orb & ")")
    Console.WriteLine("La distance entre A et B : " & D)
    Console.WriteLine("Le milieu de A et B : " & D / 2)

End Sub
```

Exercice 79

```
Dim i, j, k, n, nombre, cube As Integer
```

```
Console.WriteLine("entrer un entier (entre 0 et 9)")
n = Console.ReadLine

Console.WriteLine(" Nombres de Armstrong : ")
If (n < 10) Then
    For i = 0 To n
        Console.WriteLine(i)
    Next
Else
    For i = 1 To n
        For j = 0 To n
            For k = 0 To n
                nombre = 100 * i + 10 * j + k
                cube = i * i * i + j * j * j + k * k * k
                If (cube = nombre) Then
                    Console.WriteLine(nombre)
                End If
            Next
        Next
    Next
End If
Console.ReadLine()
```

Exercice 80

```
Dim Am() As Integer = {0, 1, 153, 371, 371, 407}
Dim test, i As Integer
Dim rep As String
Dim trouve As Boolean
Do

    trouve = False
    Console.WriteLine("veuillez saisir un entier ")
    test = Console.ReadLine()

    i = 0
    Do
        If (test = Am(i)) Then

            trouve = True

        End If
        i = i + 1
    Loop Until (trouve = True)

    If (trouve = True) Then
        Console.WriteLine("le nombre " & test & " est AMSTRONG ")
    Else
        Console.WriteLine("le nombre " & test & " est NON-AMSTRONG")
    End If
    Console.WriteLine("Voulez-vous continuer (oui/non)")
    rep = Console.ReadLine
Loop Until (rep = "non")
Console.ReadLine()
```