Année universitaire : 2022/2023

Filière Cycle d'Ingénieur « Génie Informatique »

Prof. Y FARHAOUI

# Série Nº1

#### Exercice 1

Ecrire un programme qui lit deux nombres, une opération arithmétique (+, -, \*, /) et affiche le résultat de l'opération.

#### Exercice 2

Les habitants de la ville XX paient l'impôt selon les règles suivantes :

- les hommes de plus de 20 ans paient l'impôt
- les femmes paient l'impôt si elles ont entre 18 et 35 ans
- les autres ne paient pas d'impôt

Le programme demandera donc l'âge et le sexe, et se prononcera donc ensuite sur le fait que l'habitant est imposable.

#### Exercice 3

Un magasin de reprographie facture 1 DH les dix premières photocopies, 0,6 DH les vingt suivantes et 0,4 DH au-delà. Ecrivez un programme qui demande à l'utilisateur le nombre de photocopies effectuées et qui affiche la facture correspondante.

#### Exercice 4

Un magasin est en période de solde. Il applique les remises suivantes : -30% si le prix initial du produit considéré est strictement inférieur à 100 DH, -40% si le prix initial est compris entre 100 et 200 DH et -50% si le prix initial est strictement supérieur à 200 DH. Ecrire un programme qui lit le prix d'un produit avant remise, affiche la remise correspondante et le prix après réduction. Par exemple si on fournit un prix égal à 150, le programme doit afficher « Prix initial: 150.00 DH, Remise: 60.00 DH, Prix final: 90.00 DH ».

#### Exercice 5

Un nombre différent de 1 est premier lorsque qu'il n'est divisible que par 1 et par lui-même. Écrivez un algorithme qui lit un nombre et détermine s'il est premier ou non.

#### Exercice 6

Lire un nombre. S'il est pair, afficher tous les nombres pairs qui lui sont inférieurs, sauf le zéro.

1) Lire 20 nombres au clavier; afficher le carré des nombres pairs.

2) Lire une série de nombres; afficher le carré des nombres pairs ainsi que le nombre total d'entrées et le nombre d'entrées paires. La saisie s'arrête lorsque l'utilisateur entre la valeur 100.

#### Exercice 8

Ecrire un programme qui affiche à l'écran le pattern suivant. Utiliser des boucles imbriquées !

Sec. 4221	· b)	000000000
a) 654321	1000 <b>€</b> 00	111111111
54321		22222222
4321		3333333
		44444
321		55555
21		6666
		777
1		88
		9

Les données à traiter sont constituées par une suite de nombres entiers tapés au clavier. Pour indiquer la fin de la suite, on met la valeur 999 (et on impose que tous les nombres de la suite soient différents de 999).

Exemple: Les résultats indiqués seront valables pour les données : 12 -4 6

Attention : on peut donner une « suite vide» en tapant seulement 999. Vérifier ce que deviennent vos solutions dans ce cas particulier.

Lire les données et afficher

Le nombre total de valeurs de la suite.

Résultat: 5

La somme des valeurs lues. Le minimum.

Résultat: 24

Résultat : -4

La somme des valeurs strictement positives.

Résultat: 28

Le minimum des valeurs strictement positives. Résultat : 4

#### Exercice 10

Écrire un programme C permettant d'afficher un triangle isocèle formé d'étoiles de N lignes (N étant

#### Nombre de lignes: 8



#### Exercice 11

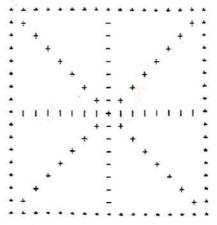
Ecrire un programme qui permet de faire des opérations sur un entier (valeur initiale à 0). Le programme affiche la valeur de l'entier puis affiche le menu suivant :

- 1. Ajouter 2
- 2. Multiplier par 3
- 3. Soustraire 5
- 4. Quitter

Le programme demande alors de taper un entier entre 1 et 4. Si l'utilisateur tape une valeur entre 1 et 3, on effectue l'opération, on affiche la nouvelle valeur de l'entier puis on réaffiche le menu et ainsi de suite jusqu'à ce qu'on tape 4. Lorsqu'on tape 4, le programme se termine.

#### Exercice 12

Ecrire un programme C qui affiche la figure suivante à l'écran (après en avoir demandé la "taille") Voilà par exemple ce que doit afficher le programme pour la taille 17. (Mais le programme doit fonctionner pour toutes les tailles!).



Année universitaire : 2022/2023 Filière Cycle d'Ingénieur « Génie Informatique »

Prof. Y FARHAOUI

# Série Nº2

# Exercice 1

Écrire un programme en C qui range au maximum (taille Nmax ) 20 nombres entiers saisis au clavier dans un tableau. Il doit gèrer en boucle le menu de choix suivant :

A- Saisie et affichage

B- Moyenne

C- Suppression du Max et affichage

D- Suppression du Min et affichage

E- Ajout d'un entier à une position donnée

Q- Quitter

Le point A sera traité par deux fonctions :

Saisie : la saisie au clavier des entiers et leur rangement dans le tableau.

Dans cette fonction on demandera le nombre d'éléments (NE<= Nmax)à saisire.

Affichage : affichage des données.

> Le point B sera traité par une fonction :

Moyenne : fonction de calcul de la moyenne du tableau (avec affichage du résultat).

> Le point C sera traité par trois fonctions :

Max\_elem : une fonction qui retourne la position du maximum des valeurs du tableau.

Supprimer: supprime le Max du tableau.

Affichage: affichage des données.

Le point **D** sera traité par trois fonctions :

Min\_elem : une fonction qui retourne la position du minimum des valeurs du tableau.

Supprimer: supprime le Mix du tableau.

Affichage: affichage des données.

Le point E sera traité par deux fonctions :

Ajout : c'est une fonction où on demande à l'utilisateur d'introduire l'entier et la

position. Puis vous insérer l'entier dans le tableau à la position indiquée.

Affichage: affichage des données.

#### Exercice 2

Déclarer et initialiser une matrice [5,5] d'entiers (iMat). Ecrire une fonction affiche matrice qui admette en paramètre une matrice [5,5] et qui imprime ses éléments sous forme de tableau. La procédure main fera un appel à affiche\_matrice pour la matrice iMat.

Déclarer un tableau iNb\_jours qui doit être initialisé de façon à ce que iNb\_jours[i] soit égal au nombre de jours du i eme mois de l'année, pour i allant de 1 à 12 (iNb\_jours[0] sera inutilisé).

Ecrire une procédure d'initialisation de iNb\_jours qui utilisera l'algorithme suivant :

si i vaut 2, le nombre de jours est de 28;

- sinon, si i est pair et i<=7 ou i impair et i>7 le nombre de jours est de 30 ;

sinon le nombre de jours est 31.

Ecrire une procédure d'impression des 12 valeurs utiles de iNb\_jours. La procédure main se contentera d'appeler les procédures d'initialisation et d'impression.

Une variante de ce programme est de réaliser une procédure d'impression pour un mois i donnée, et de faire saisir ce mois par l'utilisation dans le programme principal.

#### Exercice 4

Déclarer et initialiser deux tableaux de caractères (cTab1 et cTab2). Puis :

- 1- Ecrire une fonction longueur\_chaine1 qui admette en paramètre un tableau de caractères se terminant par un null, est qui rende le nombre de caractères du tableau (null exclu).
- 2- La procédure main imprimera le nombre d'éléments de cTab1 et cTab2 par un appel de longueur chaine1

#### Exercice 5

Algorithme de codage : on choisit un décodage (par exemple 5), et un a sera remplacé par un f, un b par un g, un c par un h, etc.. On ne crypter que les lettres majuscules et minuscules sans touches ni à la ponctuation, ni à la mise en page (caractère blancs, etc.). On suppose que les codes des lettres se suivant de a à z et de A à Z.

- 1. Déclarer un tableau de caractères cMessage initialisé avec le message en clair.
- 2. Ecrire une procédure crypt de cryptage d'un caractère qui sera passé par adresse.
- 3. Ecrire le programme principal main qui activera crypt sur l'ensemble du message et imprime le résultat.

#### Exercice 6

Ecrire un programme qui calcule le nombre de chiffres, de caractères d'espacement (espace, tabulation, fin de ligne) et tout autres caractère qu'il reçoit en entré.

# Exemple:

#### Exercice 7

Ecrire un programme se comportant comme une calculatrice, c'est-à-dire exécutant une boucle sur:

- 1. Lecture d'une ligne supposée contenir un entier, un opérateur et un entier (ex : 2 + 5). Les opérateurs sont +, -, \*, / et %.
- 2. Calcule de la valeur de l'expression. (Appel de la bonne fonction)

```
    Affichage du résultat à l'écran
    Propose le choix de recommencer ('O' ou 'N')
    Ajouter les fonctions suivantes :

            int addition ( int A, int B );
            int soustraction ( int A, int B );
            int multiplication ( int A, int B );
```

int modulo (int A, int B);

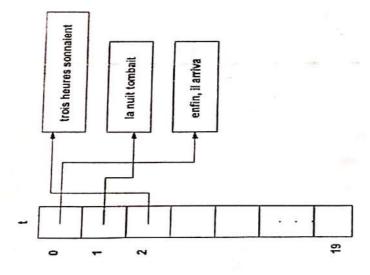
Modifier le code du programme principal afin d'appeler ces fonctions.

#### Exercice 8

Lire à partir du clavier une suite de noms ne comportant pas de caractères blancs et affichera le nombre de noms ayant plus de dix caractères. La suite se termine lorsque l'on frappe le mot « fin ». le programme doit comporter une fonction de lecture d'un nom ainsi qu'une fonction de comparaison déterminant si un nom comporte plus de dix caractères.

Vous définirez un tableau de 20 caractères dans le programme principal destiné à recevoir le nom lu au clavier. On supposera que chaque nom ne dépasse pas 20 caractères (vous pouvez inclure un test de vérification).

Vous utiliserez les fonctions de traitement des chaines de caractères dont les caractères dont le rappel de leur définition est donné ci-après :



- fonction qui compare deux chaines s et t;
- la valeur retour est positive si s est alphabétiquement supérieure à t;
- négative si s est alphabétiquement inférieure à t, int strcmp(char\* s, char\* t)

#### Exercice 9

Ecrire un programme permettant d'effectuer un ensemble d'opérations sur une chaîne de caractères quelconque saisie à partir du clavier. Ce programme est constitué d'un menu

comportant le choix de l'opération à effectuer. Les opérations sur cette chaîne sont les suivants:

Fonction saisir : elle lit une chaine de caractères à partir du clavier, elle retourne cette chaîne.

Fonction afficher: elle affiche la chaîne argument.

- Fonction inverse : elle inverse la chaine argument (elle la modifie donc)

Fonction mots : elle compte le nombre de mots de la chaine. On considère le caractère ' ' (blanc) comme le caractère séparateur de mots. Il peut y avoir plusieurs blancs consécutifs dans la chaine.

Après chaque opération, le retour au menu s'effectue après l'affichage du message 'frapper une touche pour revenir au menu'.

#### Exercice 10

1- La distance de Hamming entre deux chaînes de caractères.

La distance de Hamming entre deux chaînes de caractères de mêmes longueurs est égale au nombre de caractères, à la même position, qui sont différents.

Exemples: - La distance de Hamming entre "sure" et "cure" est 1.

- La distance de Hamming entre "aabbcc" et "xaybzc" est 3.
- Ecrire une fonction d'entête : distanceH(S1, S2) qui calcule et retourne la distance de Hamming entre S1 et S2 (les paramètres S1 et S2 sont deux chaînes de caractères de même longueur).
- 2- La distance de Hamming d'un langage.

On appellera langage, un tableau de chaîne de caractères toutes de même longueur. La distance de Hamming d'un langage est égale au minimum des distances de Hamming entre deux chaînes de caractères de ce langage différentes deux à deux.

# Exemple:

Si langage = ["aabb", "xayy", "tghy", "xgyy"], sa distance de est égale à 1.

- 🕿 Ecrire une fonction d'entête : distanceH\_langage(langage) qui calcule et retourne la distance de Hamming de son paramètre langage (le paramètre langage est un tableau de chaînes de caractères toutes de même longueur)
  - 3- La distance de Hamming entre 2 nombres entiers positifs.

La distance de Hamming entre 2 nombres entiers positifs est le nombre de bits distincts dans leurs représentations binaire (voir exemple)

# Exemple:

La distance de Hamming entre les nombres 7 et 4 est 2 (7 est représenté en binaire sur un octet (8bits) par 00000111 et 4 est représenté en binaire par 00000100)

- a- Ecrire une fonction d'entête: binaire(N) qui retourne une chaîne de caractères représentant la valeur binaire de N sur un octet ( on suppose que 0<=N<256).
- b- Ecrire une fonction d'entête: distanceNombre(A, B) qui calcule la distance de Hamming entre les nombres A et B (on suppose que 0<=A<256 et 0<=B<256)

Année universitaire: 2022/2023

Filière Cycle d'Ingénieur « Génie Informatique »

Prof. Y FARHAOUI

# Série Nº3

#### Exercice 1

Ecrire un programme qui demande un nom et affiche le nom inversé. On utilisera des pointeurs de caractères à la place du tableau de caractères. Le programme allouera dynamiquement une chaîne de 30 caractères, et emploiera deux pointeurs de caractères pour réaliser l'inversion. Le premier pointeur progressera du début vers la fin de la chaîne, et le second de la fin vers le début. Le programme doit inverser les caractères tant que les deux pointeurs ne se sont pas rencontrés.

#### Exercice 2

Programmer un détecteur de palindrome sans utiliser les fonctions de <string>. On se fondera sur l'exercice précédent. Un palindrome est un mot, un ensemble de lettres ou une phrase qui peut se lire dans les deux sens, de gauche à droite comme de droite à gauche. La langue française comprend autour de 80 mots de ce type, comme RADAR, ERE, ou REVER.

#### Exercice 3

Soit le tableau T déclaré ainsi : float t[3] [4]; Écrire les (seules) instructions permettant de calculer, la somme des éléments du tableau T:

- a. en utilisant le « formalisme usuel des tableaux à deux indices » ;
- b. en utilisant le « formalisme pointeur ».

#### Exercice 4

Soit la déclaration d'un tableau int b[3][5];

En considérant que l'allocation du tableau se fait linéairement en mémoire (les 3 « tranches » de **b** sont allouées à des adresses contiguës), donnez l'état du tableau **b** après l'exécution du code C suivant :

- / int b[3][5];
- 2 int \*a = \*b, i;
- 3 for (i = 0; i < 15; \*a++ = i++);

4 \*\*
$$b = 15$$
; 5 \*\* $(b+1) = 16$ ; 6 \* $(b[0]+1) = 17$ ; 4 \* $(*b+8) = 18$ ; 8 \* $(b[1]+2) = 19$ ; 9 \* $(*(b+1)+5) = 20$ ; 4 \* $(b[2]+3) = 21$ ; 44 \* $(*(b+2)+2) = 22$ ;

#### Exercice 5

Ecrire une fonction qui retourne la fréquence d'occurrence d'un mot dans une phrase.

#### Exercice 6

Ecrire une procédure qui enlève tous les blancs au début d'une phrase et qui laisse un seul blanc entre les différents mots de la phrase.

Écrire une fonction search qui recherche une valeur particulière dans un tableau de flottants (float). La fonction prendra en paramètre le tableau, sa taille et la valeur à rechercher. Elle retourna une valeur entière qui sera une position de la valeur dans le tableau si elle est présente dans le tableau et -1 sinon. Inclure cette fonction dans un programme qui saisit les valeurs du tableau et la valeur à rechercher.

# Exercice 8 (Devoir)

Le run-length encoding, appelé en français le codage par plages, est un algorithme de compression de données en informatique.

Le système s'applique essentiellement à des documents scannés en noir et blanc : au lieu de coder un bit par point, on dispose d'un compteur indiquant combien de points blancs ou noirs se suivent. Comme il est rare de ne pas avoir un grand nombre de pixels noirs ou de pixels blancs qui se suivent, le système a bien pour effet une compression.

Par exemple, considérons un écran de texte noir sur fond blanc. Il sera constitué de longues séquences de pixels blancs pour le fond, et de courtes séquences de pixels noirs pour le texte.

Représentons une ligne d'un tel écran, avec B pour les pixels noirs et W pour les pixels blancs

# WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW WWWWWWWWWWWWWWWWWWWWWWWWWWW

Un encodage RLE consiste alors à indiquer pour chaque suite de pixels d'une même couleur. le nombre de pixels de cette séquence. Le résultat comporte en général moins de caractères. bien que ce ne soit pas une obligation. On obtient par exemple pour la ligne précédente :

### 12W1B14W3B23W1B11W

Nous allons utiliser ce principe pour compresser un tableau d'entiers comprenant un grand nombre d'entrées redondantes. Dans notre système de compression, le premier entier indiquera le nombre d'occurrences de la valeur qui le suit. La valeur 0 sera utilisée pour compléter le tableau.

Exemple : Le codage de 4444433322110 sera représenté par : 5433222110

Année universitaire: 2022/2023

Filière Cycle d'Ingénieur « Génie Informatique »

Prof. Y FARHAOUI

## Série Nº4

#### **Exercice 1**

- 1. Définir la structure permettant de représenter un nombre complexe.
- 2. Définir les fonctions complexes suivantes (passages par valeur) :

double imag(Complex z), double real (Complex z), Complex mul(Complex z1, Complex z2)

3. Réécrire ces fonctions en utilisant le passage par adresses.

#### Exercice 2

Soient deux modèles de structure date et employe déclarés ainsi :

```
struct date {
    int jour; char mois[10]; int annee;
};
struct employe {
    char nom[15], prenom[15]
    struct date date_ naissance, date_embauche;
};
```

# Ecrire un programme qui :

1. Lit des informations pour 4 employés.

2. Affiche les employés sous la forme suivante :

Nom: Maghribi

Prenom: Mohammed

Date de naissance : 21 Janvier 1992 Date d'embauche : 05 Octobre 2010

#### Exercice 3

Soient le modèle de structure etudiant déclarés ainsi :

```
struct etudiant {
    char nom[15], prenom[15];
    int CNE;
    float notes[4], moyenne;
};
```

# Ecrire un programme qui :

- 1. Déclare un tableau d'étudiants T.
- 2. Lit les informations de 5 étudiants.
- 3. Cherche l'étudiant ayant la plus grande moyenne, puis affiche ses informations.
- 4. Trie le tableau de structures T en ordre décroissant selon la moyenne.

#### **Exercice 4**

Ecrire un programme de saisie de données pour un répertoire (nom, prénom et téléphone). Ces données doivent être placées dans un tableau de structures, chacune d'elles contenant un enregistrement. Le programme devra contenir une fonction d'affichage de toutes les données.

#### Exercice 5

Écrire une fonction: int nb\_occurrence(char T[],int n, char c), qui étant donnés un tableau T de n caractères et un caractère c, détermine les valeurs p\_occ et d\_occ qui désignent respectivement les

indices de la première et de la dernière occurrence du caractère c dans le tableau T. la fonction retourne le nombre d'occurence. 2. Écrire une fonction en langage C: int pos\_car(char T[],int n, char c, int \*p\_occ, int \*d\_occ), qui étant donnés un tableau T de n caractères et un caractère c, détermine les valeurs p\_occ et d\_occ qui désignent respectivement les indices de la première et de la dernière occurrence du caractère c dans le tableau T. la fonction retourne le nombre d'occurence.

Exemple: char c='a', T[]="baaabacaab"; donc la fonction retourne 6 avec p\_occ=1 et d\_occ=8.

#### Exercice 6

Une menuiserie industrielle gère un stock de panneaux de bois. Chaque panneau possède une largeur, une longueur et une épaisseur en millimètres, ainsi que le type de bois qui peut être Pin (code 0), chêne (code 1) ou hêtre (code 2).

- a) Définir une structure Panneau contenant toutes les informations relatives à un panneau de
- b) Ecrire des fonctions de saisie et d'affichage d'un panneau de bois.
- c) Ecrire une fonction qui calcule le volume en mètres cube d'un panneau.

#### Exercice 7

Un grossiste en composants électroniques vend quatre types de produits :

- Des cartes mères (code 1):
- Des processeurs (code 2);
- Des barrettes mémoire (code 3);
- Des cartes graphiques (code 4).

Chaque produit possède une référence (qui est un nombre entier), un prix en DH et des quantités disponibles.

- a) Définir une structure **Produit** qui code un produit.
- b) Écrire une fonction de saisie et d'affichage des données d'un produit.
- c) Écrire une fonction qui permet à un utilisateur de saisir une commande d'un produit. L'utilisateur saisit les quantités commandées et les données du produit. L'ordinateur affiche toutes les données de la commande, y compris le prix.

Considérons la structure TypeTableau suivante qui contient l'adresse d'un tableau et le nombre d'éléments du tableau.

typedef struct{ int nb elem; /\* nombre d'éléments \*/ /\* tableau \*/ int \*tab; }TypeTableau;

a) Écrire les fonctions suivantes:

TypeTableau CreationTableau(int n); qui crée un tableau de n éléments.

void DestructionTableau(TypeTableau T); qui libère la mémoire occupée par un tableau.

void SimpleLectureTableau(TypeTableau T); qui lit les éléments d'un tableau au clavier. On supposera dans cette fonction que le tableau a déjà été alloué précédemment.

void Affichage(TypeTableau T); qui affiche le contenu d'un tableau.

TypeTableau DoubleTableau(TypeTableau T); qui crée un nouveau tableau de même taille que T mais dont les éléments sont le double des éléments de T.

b) Écrire un programme principal qui saisit un tableau au clavier, calcule le double de chaque élément du tableau et affiche les résultats.

Année universitaire : 2022/2023 Filière Cycle d'Ingénieur « Génie Informatique » Prof. Y FARHAOUI

## Série N°5

(Fonctions récursives, les fichiers, Les Algorithmes de tri et de recherche)

#### Exercice 1

Écrire une fonction récursive pour calculer:

a) 
$$u_n = 1 + 2 + 3 + ... + n$$

c) 
$$u_n = 1 + 2^4 + 3^4 + \dots + n^4$$

d) 
$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}$$

e) 
$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

#### Exercice 2

Les coefficients binomiaux  $C_n^k$  pour  $k \le n$  sont donnés par la formule de Pascal :

$$C_n^0 = 1; C_n^n = 1$$

a) Ecrire une fonction récursive factoriel pour calculer le factoriel d'un entier n.

$$C_n^p = C_{n-1}^p + C_{n-1}^{p-1}$$

 $C_n^p = C_{n-1}^p + C_{n-1}^{p-1}$ b) Proposer une fonction récursive pour calculer les coefficients binomiaux.

Observer les états successifs de la mémoire pour le calcul de  $C_5^3$ .

c) Proposer un algorithme itératif pour calculer les coefficients binomiaux.

$C_0^0$	C1			
$C_2^0$	$C_2^1$	$C_2^2$		
$C_3^0$	$C_3^1$	$C_3^2$	$C_3^3$	
$C_4^0$	$C_4^1$	$C_4^2$	$C_4^3$	C4

Exercice 3 (Suite de Fibonacci)

1. Calculez le n-ième nombre de la suite de Fibonacci Fn qui est définie de la manière suivante:

$$F_0 = 0$$
,  $F_1 = 1$  et  $F_i = F_{i-1} + F_{i-2}$  pour  $i \ge 2$ .

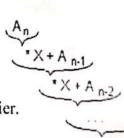
- 2. La méthode récursive est très lente pour n un peu grand car la même valeur est recalculée plusieurs fois. Proposer une variante non récursive qui stocke les valeurs dans un tableau.
- 3. Optimisez votre programme pour qu'il n'utilise que quelques entiers (un nombre indépendant de n).

#### Exercice 4

Calculer pour une valeur X donnée du type réel la valeur numérique d'un polynôme de degré n:

 $P(X) = A_n X^n + A_{n-1} X^{n-1} + ... + A_1 X + A_n$ 

Les valeurs de n, des coefficients An, ..., A0 et de X seront entrées au clavier.



# Utiliser le schéma de Horner qui évite les opérations d'exponentiation lors du calcul:

Ecrire une fonction récursive qui calcule la valeur de la fonction d'Ackermann « A » définie pour m>0 et n>0 par :

A(m, n)=A((m-1),A(m,n-1)) pour n>0, m>0;

A(0,n)=n+1 pour n>0;

A(m,0)=A(m-1,1) pour m>0;

#### Exercice 6

Ecrire une fonction récursive pgcd(m, n) qui calcule le plus grand diviseur commun des deux entiers (non-négatifs) m et n.

#### Exercice 7

Créer un fichier texte"ex6.txt"sous l'éditeur « Bloc-notes » dont les informations sont :

nom prénom age Alaoui Ali 23 Fahmi Naima 20 Hassani Hassan 35

Ecrire un programme qui crée un fichier «copie.txt», et fait la copie du fichier "ex6.txt". Tester le programme en vérifiant après exécution la présence du fichier copié dans le répertoire courant. Calculer et afficher le nombre de caractères du fichier.

Ajouter un nouvel enregistrement (entré au clavier) à la fin de "ex6.txt"et sauver le nouveau fichier sous le nom "ajou.txt".

Supprimer l'enregistrement « Hassani » du fichier "ex6.txt", puis afficher son contenu.

#### Exercice 8

Ecrire un programme C qui lit les informations contenues dans le fichier data.txt et les affiche à l'écran.

Le fichier data.txt contient les données :

Nom	Prénom	Examen1	Examen2
MALKI	Mohamed	10	11
MARWAN	Mustafa	11.5	15.5
***	220		

Notez bien que vous devez trouver un moyen pour mémoriser les données.

Dans la deuxième partie du programme, Ecrire dans un autre fichier de nom « NotesFinales.txt » les informations ci-dessous :

Nom	Prénom	Moyenne	Validation
Malki	Mohamed	10,5	NV
MARWAN	Mustafa	13.5	v

Ecrire une fonction compte\_c(FILE \* f) qui renvoie le nombre de caractères d'un fichier. Ecrire une fonction compte\_m(FILE \* F) qui renvoie le nombre de mots d'un fichier. Les mots sont séparés par des espaces ou des retours à la ligne. Ecrire une fonction compte l qui renvoie le nombre de lignes.

#### Exercice 9

Ecrire un programme C qui lit 10 valeurs entières d'un fichier d'entiers, les place dans un tableau, trie le tableau dans l'ordre croissant et place ce tableau trié dans le même fichier à la place des 10 valeurs de départ

#### Exercice 10

Soit un fichier nommé « c:\fichier.data » de données structuré en suite de lignes contenant chacune un nom de personne, un nom d'article, un nombre, et un prix. Exemple :

Alaoui assiette 10 1000

Salhi Brosse 20 567

Ecrire un programme C qui permet de lire le fichier ligne par ligne et affiche sur l'écran le contenu de chaque ligne.

#### Exercice 11

Soit un fichier de données identique à celui de l'exercice précédent. Ecrire un programme qui: 1. Permet de lire le fichier en mémorisant son contenu dans un tableau de structures, chaque structure permettent de mémoriser le contenu d'une ligne

2. Ensuite il permet de parcourir ce tableau en imprimant le contenu de chaque structure.

#### Exercice 12

Soit un fichier d'articles de commerce ayant pour structure

déclaré comme suit:

```
FILE *farticle;
```

Créer ce fichier et le remplir avec des données de votre choix.

1- Soient F1 et F2 deux fichiers d'entiers strictement positifs et sans répétition. Ecrire un programme qui construit (crée) un fichier G d'entiers tel que G contient pour chaque valeur de F1 la valeur et tous ses multiples appartenant à F2 (F1 et F2 sont supposés existants).

### Exemple:

F1:3102017

F2:3619604030

G: 3 3 6 60 30 10 60 40 30 20 60 40 17

2- Ecrire un algorithme qui permet à partir du fichier résultat (G) de générer un autre fichier (H) contenant toutes les valeurs du fichier (G) (sans répétition) avec leur nombre.

### Exemple:

H: 3 2 6 1 60 3 30 2 10 1 40 2 20 1 17 1

#### Exercice 14

Soit F un fichier d'entiers représentant des séquences de nombres séparées par un ou plusieurs zéro. Ecrire un programme qui réalise les traitements suivants :

- 1- A partir de F (fichier existant), crée un fichier G contenant pour chaque séquence, la moyenne des nombres qui la constituent.
- 2- Puis, Supprimer les valeurs nulles du fichier G.

### Exemple:

F:0014370006-927-60-10300

G: 3,75 0,00 -3,50 Avant suppression

G: 3,75 -3,50 Après suppression

Année universitaire : 2022/2023 Filière Cycle d'Ingénieur « Génie Informatique » Prof. Y FARHAOUI

# Série Nº6

(Pile, File, Arbre, Graphe)

Exercice 1: Pile (LIFO: last in, first out)

Un fichier texte peut contenir des parenthèses (), des crochets [], et des accolades {}. Ces éléments peuvent être imbriqués les uns dans les autres (exemple :  $\{a(bc[d])[\{e f\}(g)]\}$ ) Écrire une fonction qui parcourt le fichier texte et détermine si le fichier est correctement parenthésé, c'est-à-dire si toutes les parenthèses, crochets, etc. sont bien refermés par un caractère du même type, et si les parenthèses, crochets et accolades sont correctement imbriqués. Exemple de fichier incorrect :  $\{\{\}\}$ .

Exercice 2: File (FIFO: first in, first out)

Dans une gare, un guichet est ouvert. Les clients arrivent à des dates aléatoires et rentrent dans une queue. L'intervalle entre l'arrivée de deux clients successifs est un nombre aléatoire entre 0 et INTERVALLE\_MAX (les dates sont des entiers indiquant des secondes). Lorsque le guichetier a fini de traiter un client, il appelle le client suivant dont le traitement va avoir une durée aléatoire entre 0 et DUREE\_TRAITEMENT\_MAX.

1. Définir les structures de données pour l'algorithme de simulation.

2. Écrire une fonction *CreerListeClients*, qui crée une file de clients, le nombre de clients étant saisi au clavier. Cette fonction initialise aussi la date d'arrivée et la durée d'attente de chacun des clients. On supposera que le premier client est arrivé à 8h.

3. Écrire une fonction d'affichage qui affiche le numéro de chacun des clients, sa date d'arrivée et sa date de fin de traitement en format (h min sec).

#### Exercice 3:

La configuration matérielle du réseau informatique intranet d'une entreprise nationale est enregistrée dans un fichier texte. Le fichier contient :

- Le nombre *n* de nœuds du réseau (chaque nœud correspondant plus ou moins à une localisation géographique);
- Le nombre *m* de connexions entre nœuds (une connexion correspondant à un câble). On supposera qu'entre deux nœuds il y a au plus 1 câble.
- La liste des n noms des nœuds ;
- La liste des m connexions, chaque connexion étant représentée par les numéros des deux nœuds extrémités.
- 1) Proposez une structure de données pour représenter le réseau en mémoire centrale. On rassemblera toutes les données nécessaires dans une structure Reseau.
- 2) Écrire une fonction de chargement du réseau en mémoire.
- 3) Écrire une fonction de sauvegarde du réseau stocké en mémoire.

4) Écrire une fonction qui détermine si deux machines données sont connectées par un câble.

5) Qu'est-ce qu'un chemin dans le réseau?

6) Écrire une fonction qui prend en paramètre un tableau de numéros de nœuds du réseau et qui détermine si chaque nœud du tableau est connecté au suivant par un câble.

# Exercice 4: (graphes non orientés)

On appelle graphe non orienté un graphe dont toutes les arêtes vont dans les deux sens. Autrement dit, dans un graphe non orienté, s'il y a un arc d'un sommet si vers le sommet s j, alors il y a aussi un arc du sommet j vers le sommet s j. La paire {i, j} est alors appelée une arête de G.

Soit un graphe donné par une matrice d'adjacence. Donner une propriété de la matrice qui soit caractéristique d'un graphe non orienté. Donner un algorithme qui prend en paramètre un graphe donné sous forme de matrice d'adjacence, et qui renvoie 1 si le graphe est non orienté, et 0 sinon.

#### Exercice 5:

Soit un graphe G à n sommets. Un coloriage de G est une fonction qui à chaque sommet de G associe un entier appelé couleur du sommet. On représente un coloriage de G par un tableau de n entiers.

Un coloriage de G est dit correct si pour tout arc (i, j) de G la couleur de si est différente de la couleur de s j.

Écrire une fonction qui prend en paramètre un graphe G représenté sous forme de matrice d'adjacence et un coloriage, et qui renvoie 1 si le coloriage de G est correct et 0 sinon.