

Devoir-surveillé <input type="checkbox"/>	Examen <input checked="" type="checkbox"/>	Session :	principale <input checked="" type="checkbox"/> de contrôle <input type="checkbox"/>
Matière : Algorithmique et structure de données		Semestre: 1	
Enseignant(s) : W. Miled & O. Mosbahi		Date: 05/01/2012 à 10h30	
Filière(s) : MPI		Durée: 1h30	
Barème indicatif: 3, 4.5, 4.5, 8		Documents: non autorisés <input checked="" type="checkbox"/>	
Nombre de pages : 2			

Remarque : On veillera au soin apporté dans la présentation et la clarté de la solution.

Exercice 1

Indiquez les erreurs se trouvant dans le programme ci-dessous et corrigez-les :

```

1. include <stdio.h>
2.
3. void main(void) {
4.     char *chaine;
5.
6.     printf("Entrez votre nom ? ");
7.     scanf("%c", &chaine);
8.
9.     printf("Votre nom est : ");
10.    for (int i = 0; i < strlen(chaine)-1; i++)
11.    {
12.        printf("%c", chaine[i]);
13.    }
14.    return 0
15. }
```

Exercice 2

Une suite qui calcule les puissances de 2 peut s'écrire :

$$U_n = 2^n = 1 \times \underbrace{2 \times 2 \times \dots \times 2 \times 2}_{n \text{ fois}} \text{ ou bien encore } \begin{cases} U_0 = 1 \\ U_n = 2 \times U_{n-1} \end{cases}$$

1. On se propose d'écrire une fonction en C qui calcule de manière itérative la valeur U_n de la suite pour une valeur de n donnée. Pour cela, on écrira les 2 versions successives suivantes :

```

int Calcul_Suite_iter1(int n).
void Calcul_Suite_iter2(int n, int * valeur).
```

2. Définir la fonction `int Calcul_Suite_rec(int n)` qui calcule de manière récursive la valeur U_n pour une valeur de n donnée.

3. Écrire une fonction `main()` qui fait saisir à l'utilisateur la valeur de n , puis qui affiche la valeur de U_n calculée de manière itérative et récursive par les trois fonctions définies précédemment.

Exercice 3

On considère un tableau à deux dimensions dont les éléments sont des entiers et tel que le nombre de lignes est égal au nombre de colonnes.

1. Ecrire une fonction en C, nommée `Remplir` qui remplit le tableau en initialisant le centre avec la valeur `x` et les autres éléments du centre vers l'extérieur avec des valeurs s'incrémentant de `p` à chaque étape.
2. Ecrire la fonction `main` qui saisit au clavier la taille `n` du tableau (tel que `n` soit un entier positif impair), et deux entiers quelconques `x` et `p` et qui fait appel à la fonction `Remplir`.

Exemple : Pour `n=7`, `p=3` et `x=2`

Initialisation

			2			

Tableau obtenu

11	11	11	11	11	11	11
11	8	8	8	8	8	11
11	8	5	5	5	8	11
11	8	5	2	5	8	11
11	8	5	5	5	8	11
11	8	8	8	8	8	11
11	11	11	11	11	11	11

Exercice 4 (Cet exercice sera traité au choix en algorithmique ou en C)

On considère un tableau à deux dimensions `M` dont les éléments sont des lettres minuscules, et tel que le nombre de lignes est égal au nombre de colonnes. Ce tableau `M` sera désormais appelé la matrice carrée `M`. On souhaite écrire un programme qui permet de dessiner l'histogramme des éléments de la matrice. Un histogramme est un digramme qui détermine le nombre de fois où chaque élément est présent dans la matrice.

Pour cela, la méthode consiste à construire deux tableaux à une dimension :

- Un tableau `Elements`
- Un tableau `Histo`

1. Ecrire une fonction nommée `Remplir_Tableau2D` qui saisit le nombre de lignes `n` et qui remplit la matrice `M` par des caractères. Seules les lettres minuscules seront mémorisées.
2. Ecrire une fonction nommée `Calcul_Elements` qui étant donnée, la matrice `M` de taille `n`, va construire le tableau `Elements` qui va contenir tous les éléments de la matrice une seule fois chacun.
3. Ecrire une fonction nommée `Calcul_Histo` qui étant donnée, la matrice `M` de taille `n`, va construire le tableau `Histo` dans lequel chaque case d'indice `i` indique le nombre de fois où le caractère `Elements[i]` est présent dans la matrice.
4. Ecrire une fonction nommée `Affiche_Histo` qui étant données les deux tableaux `Elements` et `Histo`, va afficher par ligne, un élément du tableau `Elements` et un nombre d'étoiles "*" correspondant au nombre présent dans le tableau `Histo`.
5. Ecrire le programme principal qui fait appel à toutes ces fonctions.

Exemple : Soit la matrice suivante :

a	b	c	a
c	c	b	a
a	b	a	c
c	a	a	a

L'histogramme obtenu est :

a : * * * * *

b : * * *

c : * * * *

`Elements` :

a	b	c
---	---	---

`Histo` :

8	3	5
---	---	---