

Questions préparatoires à l'examen No 1

Question 1

Écrire un programme qui lit une séquence de nombres entiers. À la première valeur négative, la lecture des données s'arrête et les informations suivantes sont affichées à l'écran : le nombre d'entiers lus, la valeur minimale et la valeur maximale dans la liste (sans inclure la valeur négative). Si la liste est vide (donc, commence immédiatement avec une valeur négative) le programme affiche 0, 0, 0.

Question 2

Définissez la fonction « nbtermes » dont le prototype est :

```
int nbtermes(long max);
```

Cette fonction doit calculer et retourner le nombre de termes nécessaires pour que la série (1, 3, 6, 10, 15, 21, 28, 36, 45, 55, ...etc) dépasse la valeur « max » reçue en paramètre. La série est définie comme la somme des valeurs de 1 à n.

Question 3

Dans réaliser ce programme, imaginons d'abord que nous nous trouvons devant une grille à la position (0,0). Le but du programme est d'effectuer des déplacements aléatoires tant et aussi longtemps que le déplacement est situé à l'intérieur d'un cercle de rayon R (qui sera fourni par l'utilisateur). À chaque coup, vous vous déplacez au hasard de la position (x, y) à une des quatre positions voisines (x+1, y), (x-1, y), (x, y+1), ou (x, y-1). Tant que $x^2 + y^2 \leq R^2$, le programme continue les déplacements. À la fin, le programme affiche le nombre de déplacements effectués. **AIDE** : Générez un nombre aléatoire entre 0 et 3 pour connaître la direction à prendre.

Question 4

Complétez la boucle qui accumule dans « tot » la somme des chiffres d'un nombre « x ». Exemples : Si « x » vaut 1234, alors « tot » vaudra 10. Si « x » vaut 10201 alors « tot » vaudra 4.

```
int x, tot = 0;

while (x) {
    _____ ;
    x /= 10 ;
}
```

Question 5

Corrigez toutes les erreurs que vous voyez dans ce bloc et expliquez :

```
#define MAX 10

void main( )
{   int a;

    scanf("%d", a);
    MAX = 2*a;
    printf("résultat = %d", MAX); }
```

Question 6

Complétez les indices des boucles FOR dans ce programme pour que le programme puisse afficher le résultat illustré ci-contre:

1	2	3	4	5
	2	3	4	5
		3	4	5
			4	5
				5

```
#include <stdio.h>

#define MAX 5

void main() {
    int I, J;

    FOR ( _____ )           (A)
    {

        FOR ( _____ )       (B)
        printf("  ");

        FOR ( _____ )       (C)
        printf("%d ", J);

        printf("\n");
    }
}
```

Question 7

Écrire la fonction « **long Fibo(int N);** » qui calcule et retourne le N^{ième} terme (pour tout N>0) de la série de fibonacci. À partir des 2 premiers termes (1, 1), la série de fibonacci est contruite en faisant la somme des 2 termes précédents (pour tout i > 2):

$$\text{terme}_i = (\text{terme}_{i-2}) + (\text{terme}_{i-1})$$

Ex. (8 premiers termes) → 1, 1, 2, 3, 5, 8, 13, 21, ...etc

Question 8

Écrire la fonction « `float PuissR(float x, int n);` » qui calcule et retourne la valeur de x^n en utilisant la méthode de calcul de puissance à la russe. Voici les étapes de cette méthode :

1. On commence avec un accumulateur « total » à 1
2. Si la valeur « n » est impaire, on multiplie « total » par « x »
3. On multiplie « x » par lui-même pour le mettre au carré, et on divise « n » par 2 (division entière!!)
4. On répète 2-3 jusqu'à ce que la valeur de « n » soit 0.

Trace d'exécution pour $x = 2$ et $n = 13$:

X	N	Total
		1
2	13	=1 * 2 = 2
4	6	=2
16	3	=2 * 16 = 32
256	1	=32 * 256 = 8192
65536	0	<u>8192</u>

Question 9

Écrire une boucle qui effectue la lecture répétitive d'un choix entier : seulement les chiffres 0, et 1 doivent être acceptés.

Question 10

Écrire une fonction `int somme_cubes(int x);` qui retourne 1 (vrai) si une valeur « x » est égale à la somme des cubes de chacun de ses chiffres, et sinon retourne 0 (faux).

Exemple: $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27$ `somme_cubes(153) → 1`

Parmi tous les entiers plus grand que 1, seulement quatre peuvent être représentés par la somme des cubes de leurs chiffres. Sachant que ces quatre nombres sont compris entre 150 et 410, écrire un programme pour trouver et afficher ces quatre nombres en utilisant « `somme_cubes()` ».

AIDE :

- l'opérateur « / » effectue une division entière. Ex. $17 / 3 \rightarrow 5$
- l'opérateur « % » retourne le reste d'une division entière. Ex. $17 \% 3 \rightarrow 2$
- la fonction « `double pow (double x, double n);` » de `<math.h>` retourne « x^n »