



Examen final

Aucun document n'est autorisé
Les solutions doivent être rédigées en C
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

1 Affichage

6 pts. ☹️20'

Qu'affiche le programme suivant :

```
1 #include<stdio.h>
2 int a =1, b = 10;
3 int c=20;
4 void func_1()
5 {
6     printf("func_1 : a=%d, b=%d\n", a, b);
7 }
8 void func_2()
9 {
10     int a = 5;
11     printf("func_2 : a=%d, b=%d\n", a, b);
12 }
13 void func_3(int b)
14 {
15     a = b;
16     printf("func_3 : a=%d, b=%d\n", a, b);
17 }
```

```
18 void main()
19 {     int i =1, j=3;
20         int m =0, n=0;
21         int a = 3;
22         printf("a=%d, b=%d\n", a, b);
23         func_1();
24         func_2();
25         func_3(b);
26         printf("a=%d, b=%d\n", a, b);
27         for(i=0; j; j-- )
28         {
29             int c;
30             i++;
31             c = i;
32         }
33         printf("i=%d, c=%d\n", i, c);
34 }
```

2 Carré Latin

6 pts. ☹️30'

Un carré latin est un tableau carré de n lignes et n colonnes remplies de n éléments distincts dont chaque ligne et chaque colonne ne contient qu'un seul exemplaire. La plupart du temps, les n éléments utilisés sont les entiers compris entre 0 et $n-1$.

```
0 1 2 3 4
1 2 3 4 0
2 3 4 0 1
3 4 0 1 2
4 0 1 2 3
```

Voici un exemple de carré latin d'ordre 5 :

Pour tout entier n , on peut construire (ligne par ligne) un carré latin d'ordre n par permutation circulaire des chiffres de 0 à $n-1$ comme suit :

```
0 1 2 ... n-1
1 2 3 ... 0
2 3 4 ... 1
⋮ ⋮ ⋮ ⋱ ⋮
n-1 0 1 ... n-2
```

Écrire un programme qui demande à l'utilisateur de saisir un entier n puis affiche le carré latin d'ordre n en utilisant le principe décrit ci-dessus.

3 Les nombres heureux

8 pts. ☹40'

En mathématiques, un entier naturel est un nombre heureux si, lorsqu'on calcule la somme des carrés de ses chiffres dans son écriture en base dix puis la somme des carrés des chiffres du nombre obtenu et ainsi de suite, on aboutit au nombre 1.

De manière plus formelle, on considère un entier positif t , puis on définit la suite d'entiers $t_0, t_1, t_2 \dots$ où $t_0 = t$ et t_{i+1} est égal à la somme des carrés des chiffres de t_i . t est dit heureux si la suite aboutit à 1 à partir d'un certain nombre de termes, c'est-à-dire que pour un certain indice i , $t_i = 1$ (à partir de cet indice, tous les t_j sont égaux à 1 et la suite est constante).

Exemple :

Le nombre 7 est heureux, puisque sa suite associée est :

$$t_1 = 7^2 = 49;$$

$$t_2 = 4^2 + 9^2 = 97$$

$$t_3 = 9^2 + 7^2 = 130$$

$$t_4 = 1^2 + 3^2 + 0^2 = 10$$

$$t_5 = 1^2 + 0^2 = 1.$$

On peut démontrer qu'en appliquant cette méthode, à partir d'un entier quelconque non nul, on finit par boucler sur un des cycles suivants : {1} si le nombre est heureux. Sinon, dès que, dans la suite associée à un nombre, on rencontre l'une des valeurs suivantes : {4, 16, 37, 58, 89, 145, 42, 20}, la suite devient donc périodique et le nombre en question est dit malheureux, puisque : $\dots, 4^2 = 16, 1^2 + 6^2 = 37, 3^2 + 7^2 = 58, 5^2 + 8^2 = 89, 8^2 + 9^2 = 145, 1^2 + 4^2 + 5^2 = 42, 4^2 + 2^2 = 20, 2^2 + 0^2 = 4, \dots$

Questions :

1. Écrire une fonction SommeChiffresCarres qui prend un entier comme paramètre et renvoie la somme des carrés de ses chiffres.
2. Écrire une fonction EstHeureux qui permet de tester si un nombre passé comme paramètre est un nombre heureux en retournant 1 (si oui), et 0 sinon.
3. Écrire une fonction AfficherNombresHeureux qui affiche tous les nombres heureux compris entre deux valeurs entières passées comme paramètres.
4. Écrire une fonction main() qui demande à l'utilisateur de saisir deux entiers puis appelle la fonction AfficherNombresHeureux pour afficher tous les nombres heureux compris entre les deux valeurs saisies.



Correction de l'examen final

Aucun document n'est autorisé
Les solutions doivent être rédigées en C
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

1 Affichage

6 pts. ☺20'

Qu'affiche le programme suivant :

```
1 #include<stdio.h>
2 int a =1, b = 10;
3 int c=20;
4 void func_1()
5 {
6     printf("func_1 : a=%d, b=%d\n", a, b);
7 }
8 void func_2()
9 {
10     int a = 5;
11     printf("func_2 : a=%d, b=%d\n", a, b);
12 }
13 void func_3(int b)
14 {
15     a = b;
16     printf("func_3 : a=%d, b=%d\n", a, b);
17 }
```

```
18 void main()
19 {     int i =1, j=3;
20         int m =0, n=0;
21         int a = 3;
22         printf("a=%d, b=%d\n", a, b);
23         func_1();
24         func_2();
25         func_3(b);
26         printf("a=%d, b=%d\n", a, b);
27         for(i=0; j; j-- )
28         {
29             int c;
30             i++;
31             c = i;
32         }
33         printf("i=%d, c=%d\n", i, c);
34 }
```

Solution

Affichage

```
a=3, b=10
func_1 : a=1, b=10
func_2 : a=5, b=10
func_3 : a=10, b=10
a=3, b=10
i=3, c=20
```

2 Carré Latin

6 pts. ☺30'

Un carré latin est un tableau carré de n lignes et n colonnes remplies de n éléments distincts dont chaque ligne et chaque colonne ne contient qu'un seul exemplaire. La plupart du temps, les n éléments utilisés sont les entiers compris entre 0 et $n - 1$.

Voici un exemple de carré latin d'ordre 5 :

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

Pour tout entier n , on peut construire (ligne par ligne) un carré latin d'ordre n par permutation circulaire des chiffres de 0 à $n - 1$ comme suit :

| | | | | |
|----------|----------|----------|----------|----------|
| 0 | 1 | 2 | ... | $n - 1$ |
| 1 | 2 | 3 | ... | 0 |
| 2 | 3 | 4 | ... | 1 |
| \vdots | \vdots | \vdots | \ddots | \vdots |
| $n - 1$ | 0 | 1 | ... | $n - 2$ |

Écrire un programme qui demande à l'utilisateur de saisir un entier n puis affiche le carré latin d'ordre n en utilisant le principe décrit ci-dessus.

Solution

```

1 #include <stdio.h>
2
3 int main()
4 {
5     int N, i, j;
6
7     printf("Donnez N: ");
8     scanf("%d", &N);
9
10    for(i=0; i<N; i++)
11    {
12        for(j=i; j<N; j++)
13        {
14            printf("%d ", j);
15        }
16
17        for(j=0; j<i; j++)
18        {
19            printf("%d ", j);
20        }
21
22        printf("\n");
23    }
24
25    return 0;
26 }
```

3 Les nombres heureux

8 pts. ☹️40'

En mathématiques, un entier naturel est un nombre heureux si, lorsqu'on calcule la somme des carrés de ses chiffres dans son écriture en base dix puis la somme des carrés des chiffres du nombre obtenu et ainsi de suite, on aboutit au nombre 1.

De manière plus formelle, on considère un entier positif t , puis on définit la suite d'entiers $t_0, t_1, t_2 \dots$ où $t_0 = t$ et t_{i+1} est égal à la somme des carrés des chiffres de t_i . t est dit heureux si la suite aboutit à 1 à partir d'un certain nombre de termes, c'est-à-dire que pour un certain indice i , $t_i = 1$ (à partir de cet indice, tous les t_j sont égaux à 1 et la suite est constante).

Exemple :

Le nombre 7 est heureux, puisque sa suite associée est :

$$t_1 = 7^2 = 49;$$

$$t_2 = 4^2 + 9^2 = 97$$

$$t_3 = 9^2 + 7^2 = 130$$

$$t_4 = 1^2 + 3^2 + 0^2 = 10$$

$$t_5 = 1^2 + 0^2 = 1.$$

On peut démontrer qu'en appliquant cette méthode, à partir d'un entier quelconque non nul, on finit par boucler sur un des cycles suivants : {1} si le nombre est heureux. Sinon, dès que, dans la suite associée à un nombre, on rencontre l'une des valeurs suivantes : {4, 16, 37, 58, 89, 145, 42, 20}, la suite devient donc périodique et le nombre en question est dit malheureux, puisque : $\dots, 4^2 = 16$, $1^2 + 6^2 = 37$, $3^2 + 7^2 = 58$, $5^2 + 8^2 = 89$, $8^2 + 9^2 = 145$, $1^2 + 4^2 + 5^2 = 42$, $4^2 + 2^2 = 20$, $2^2 + 0^2 = 4, \dots$

Questions :

1. Écrire une fonction SommeChiffresCarres qui prend un entier comme paramètre et renvoie la somme des carrés de ses chiffres.

Solution

```
1 int SommeChiffresCarres(int num){
2     int chiffre = 0, sum = 0;
3     while(num > 0){
4         chiffre = num%10;
5         sum = sum + (chiffre*chiffre);
6         num = num/10;
7     }
8     return sum;
9 }
```

2. Écrire une fonction EstHeureux qui permet de tester si un nombre passé comme paramètre est un nombre heureux en retournant 1 (si oui), et 0 sinon.

Solution

```
1 int EstHeureux(int num){
2     int t = num;
3     do{
4         t = SommeChiffresCarres(t);
5         if(t == 4) return 0;
6     }while(t != 1);
7     return 1;
8 }
```

3. Écrire une fonction AfficherNombresHeureux qui affiche tous les nombres heureux compris entre deux valeurs entières passées comme paramètres.

Solution

```
1 void AfficherNombresHeureux(long n, long m)
2 {
3     while(n <= m)
4     {
5         if(EstHeureux(n))
6         {
7             printf("%d, ", n);
8         }
9         n++;
10    }
11 }
```

4. Écrire une fonction main() qui demande à l'utilisateur de saisir deux entiers puis appelle la fonction AfficherNombresHeureux pour afficher tous les nombres heureux compris entre les deux valeurs saisies.

Solution

```
1 int main()
2 {
3     long n, m;
4     printf("Donnez la premiere valeur n : ");
5     scanf("%d", &n);
6     printf("Donnez la deuxieme valeur m : ");
7     scanf("%d", &m);
8
9     printf("Tous les nombres heureux entre %d et %d sont : \n", n, m);
10    AfficherNombresHeureux(n, m);
11
12    return 0;
13 }
```