



Examen Final

Aucun document n'est autorisé.

*Les solutions peuvent être rédigées en C ou en langage algorithmique.
Tout appareil électronique doit être éteint (Téléphone, Ordinateur, Tablette, etc.).*

1 Schtroumpf

(7 pts, 15 min)

Qu'affiche le programme suivant :

```
1 #include <stdio.h>
2 int c = 0 ;
3 void schtroumpf (int a, int b)
4 {
5     printf ("Debut de schtroumpf : a=%d et b=%d\n", a, b) ;
6     c = a ;
7     a = b ;
8     b = c ;
9     printf ("Fin de schtroumpf : a=%d et b=%d\n", a, b) ;
10 }
11
12 void main()
13 {
14     int n=4, p=8 ;
15     c=n+p;
16     printf ("Avant l'appel : n=%d et p=%d\n", n, p) ;
17     schtroumpf (n, p) ;
18     printf ("Après l'appel : n=%d et p=%d\n", n, p);
19     printf ("c=%d", c);
20 }
```

2 Nombre palindrome

(4 pts, 35 min)

Un nombre palindrome est un nombre symétrique, *c-à-d* égal à lui-même s'il est lu de gauche à droite ou de droite à gauche. Par exemple : 11, 636, 42924 et 85077058 sont des nombres palindromes.

Écrire un programme qui demande à l'utilisateur un entier puis affiche si ce dernier est palindrome ou non.

3 Amitié numérique

(9 pts, 40 min)

Deux nombres M et N sont appelés nombres amis si la somme des diviseurs de M est égale à N et la somme des diviseurs de N est égale à M

1. Écrire une fonction `sommeDiviseurs` qui retourne la somme des diviseurs d'un nombre passé en paramètre.
2. Écrire une fonction `monAmi` qui retourne le nombre ami (s'il existe) d'un nombre passé en paramètre.
3. Écrire un programme principal qui affiche tous les nombres amis inférieurs à une certaine limite donnée par l'utilisateur.



Correction de l'examen final

Aucun document n'est autorisé.

Les solutions peuvent être rédigées en C ou en langage algorithmique.

Tout appareil électronique doit être éteint (Téléphone, Ordinateur, Tablette, etc.).

1 Schtroumpf

(7 pts, 15 min)

Qu'affiche le programme suivant :

```
1 #include <stdio.h>
2 int c = 0 ;
3 void schtroumpf (int a, int b)
4 {
5     printf ("Debut de schtroumpf : a=%d et b=%d\n", a, b) ;
6     c = a ;
7     a = b ;
8     b = c ;
9     printf ("Fin de schtroumpf : a=%d et b=%d\n", a, b) ;
10 }
11
12 void main()
13 {
14     int n=4, p=8 ;
15     c=n+p;
16     printf ("Avant l'appel : n=%d et p=%d\n", n, p) ;
17     schtroumpf (n, p) ;
18     printf ("Après l'appel : n=%d et p=%d\n", n, p);
19     printf ("c=%d", c);
20 }
```

Solution

```
Avant l'appel : n=4 et p=8
Debut de schtroumpf : a=4 et b=8
Fin de schtroumpf : a=8 et b=4
Après l'appel : n=4 et p=8
c=4
Process returned 3 (0x3)   execution time : 0.020 s
Press any key to continue.
```

2 Nombre palindrome

(4 pts, 35 min)

Un nombre palindrome est un nombre symétrique, *c-à-d* égal à lui-même s'il est lu de gauche à droite ou de droite à gauche. Par exemple : 11, 636, 42924 et 85077058 sont des nombres palindromes.

Écrire un programme qui demande à l'utilisateur un entier puis affiche si ce dernier est palindrome ou non.

Solution

```

1 #include<stdio.h>
2
3 main()
4 {
5     int nombre,inverse,nbr=0;
6     printf("Entrez votre nombre: ");
7     scanf("%d", &nombre);
8     inverse=0;
9     nbr=nombre;
10    do
11    {
12        inverse=(inverse*10)+ nbr%10;
13        nbr=nbr/10;
14    }while (nbr>0);
15
16    if(nombre==inverse)
17        printf("%d est un palindrome", nombre);
18    else
19        printf("%d n'est pas un palindrome", nombre);
20 }

```

3 Amitié numérique

(9 pts, 40 min)

Deux nombres M et N sont appelés nombres amis si la somme des diviseurs de M est égale à N et la somme des diviseurs de N est égale à M

1. Écrire une fonction `sommeDiviseurs` qui retourne la somme des diviseurs d'un nombre passé en paramètre.

Solution

```

1 int sommeDiviseurs ( int nombre )
2 {
3     int i, somme = 0;
4     for (i=1; i<=nombre/2; i++)
5     {
6         if (nombre % i == 0)
7             somme = somme + i;
8     }
9     return somme;
10 }

```

2. Écrire une fonction `monAmi` qui retourne le nombre ami (s'il existe) d'un nombre passé en paramètre.

Solution

```

1 int monAmi ( int nombre )
2 {
3     int unAmis;
4     unAmis = sommeDiviseurs(nombre);
5     if (sommeDiviseurs(unAmis) == nombre)
6         return unAmis;
7     else
8         return nombre;
9 }

```

3. Écrire un programme principal qui affiche tous les nombres amis inférieurs à une certaine limite donnée par l'utilisateur.

Solution

```
1 #include <stdio.h>
2 /* calcul de la somme des diviseurs du parametre nombre */
3 int sommeDiviseurs ( int nombre );
4 /* calcul de l'ami du parametre nombre */
5 int monAmi ( int nombre );
6
7 int main ( )
8 {
9     int nombre, limite, nombre_ami;
10    printf("Limite : "); scanf("%d", &limite);
11    for (nombre=1; nombre<=limite; nombre++)
12    {
13        nombre_ami = monAmi(nombre);
14        if (nombre_ami != nombre)
15            printf("%d et %d sont amis\n", nombre, nombre_ami);
16    }
17    return 0;
18 }
```