



Épreuve de Rattrapage

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

☑☑ pts. ⌚20'

Qu'affiche les deux programmes suivants :

```
1 void Toto(int a)
2 {
3   printf("%d \n", a);
4   a = a+1;
5   printf("%d \n", a);
6 }
7
8 main()
9 {
10  int b;
11  b=2;
12  printf("%d \n", b);
13  Toto(b+1);
14  printf("%d \n", b*3);
15  return 0;
16 }
```

```
1 void Loulou(int X, int *Y)
2 {
3   X = X+1;
4   *Y = *Y+1;
5   printf("%d %d \n", X, *Y);
6 }
7
8 main()
9 {
10  int A,B;
11  A=0;
12  B=0;
13  Loulou(A, &B);
14  printf("%d %d \n", A, B);
15  return 0;
16 }
```

2 Triangle isocèle

☑ pts. ⌚30'

Écrire une fonction qui prend en entrée un entier N puis affiche un triangle isocèle formé d'étoiles de N lignes. Écrire un programme pour tester la fonction.

Par exemple : pour N = 6

```
      *
     ***
    *****
   ********
  *********
 *****
*****
```

3 Nombre factorion

☑ pts. ⌚40'

Écrire une fonction qui détermine si un nombre entier positif donné comme paramètre est un factorion. Écrire un programme pour tester la fonction.

Un factorion est un nombre entier positif qui est égal à la somme des factorielles de ses chiffres.

Exemples :

$$145 = 1! + 4! + 5!$$

$$40585 = 4! + 0! + 5! + 8! + 5!$$

Bon Courage !



Correction de l'épreuve de Rattrapage

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

☑☑ pts. ⌚20'

Qu'affiche les deux programmes suivants :

```
1 void Toto(int a)
2 {
3   printf("%d \n", a);
4   a = a+1;
5   printf("%d \n", a);
6 }
7
8 main()
9 {
10  int b;
11  b=2;
12  printf("%d \n", b);
13  Toto(b+1);
14  printf("%d \n", b*3);
15  return 0;
16 }
```

```
1 void Loulou(int X, int *Y)
2 {
3   X = X+1;
4   *Y = *Y+1;
5   printf("%d %d \n", X, *Y);
6 }
7
8 main()
9 {
10  int A,B;
11  A=0;
12  B=0;
13  Loulou(A, &B);
14  printf("%d %d \n", A, B);
15  return 0;
16 }
```

Solution



2 Triangle isocèle

☑ pts. ⌚30'

Écrire une fonction qui prend en entrée un entier N puis affiche un triangle isocèle formé d'étoiles de N lignes. Écrire un programme pour tester la fonction.

Par exemple : pour N = 6

Bon Courage !

```
*
***
****
*****
*****
*****
*****
```

Solution

```
1 #include <stdio.h>
2 void affichage_triangle(int LIG)
3 {
4     int L;      /* compteur des lignes      */
5     int ESP;    /* nombre d'espaces      */
6     int I;      /* compteur des caractères */
7     for (L=0 ; L<LIG-1 ; L++)
8     {
9         ESP = LIG-L;
10        for (I=0 ; I<ESP ; I++)
11            putchar(' ');
12        for (I=0 ; I<2*L+1 ; I++)
13            putchar('*');
14
15        putchar('\n');
16    }
17 }
18 main()
19 {
20     int LIG;    /* nombre de lignes      */
21
22     do
23     {
24         printf("Nombres de lignes : ");
25         scanf("%d", &LIG);
26     }
27     while (LIG<1 || LIG>20);
28     affichage_triangle(LIG);
29
30     return 0;
31 }
```

3 Nombre factorion

📄 pts. ⌚40'

Écrire une fonction qui détermine si un nombre entier positif donné comme paramètre est un factorion. Écrire un programme pour tester la fonction.

Un factorion est un nombre entier positif qui est égal à la somme des factorielles de ses chiffres.

Exemples :

$$145 = 1! + 4! + 5!$$

$$40585 = 4! + 0! + 5! + 8! + 5!$$

Solution

Bon Courage !

```

1 #include <stdio.h>
2 int factorion(int n)
3 {
4     int m = n;
5     int somfact = 0;
6     while(m>0){
7         int chiffre = m%10;
8         int fact = 1;
9         int i;
10        for (i = 2; i<=chiffre ; ++i)
11            fact *= i;
12        somfact += fact;
13        m /= 10;
14    }
15
16    return (somfact == n);
17 }
18 void main(void)
19 {
20     int n;
21     printf("donnez un entier positif n : ");
22     scanf("%d",&n);
23     if (factorion(n)) printf("%d est un factorion.\n", n);
24     else printf("%d n'est pas un factorion.\n", n);
25 }

```

Bon Courage !