



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. ⌚30'

1. Qu'affichent les deux programmes suivants :

```
1 /////////////// Programme 1 ///////////////
2 int main(){
3     int num1 = 1, num2 = 2;
4     switch(num1 > num2)
5     {
6         case 0:
7             printf("%d", num2); break;
8         case 1:
9             printf("%d", num1); break;
10        default:
11            printf("%d", num1+num2); break;
12    }
13 }
```

```
1 /////////////// Programme 2 ///////////////
2 #include <stdio.h>
3 int main()
4 {
5     int num = 123, toto = 0;
6     while(num != 0)
7     {
8         toto = (toto*10)+(num%10);
9         num /= 10;
10    }
11    printf("toto = %d", toto);
12    return 0;
13 }
```

2. Que font les deux programmes Programme 1 et Programme 2? (deux lignes au maximum pour chacun d'entre eux.)

2 Affichage de chiffres

☑ pts. ⌚30'

Écrire un programme qui demande à l'utilisateur un entier N positif puis affiche le motif suivant avec les chiffres allant de 1 à N..

Par exemple : N = 5

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

3 Nombre d'Armstrong

☑ pts. ⌚30'

Écrire une fonction qui détermine si un nombre entier positif donné comme paramètre est un nombre d'Armstrong, puis écrire un programme pour tester cette fonction.

On rappelle qu'un nombre d'Armstrong est un entier naturel qui est égal à la somme des cubes des chiffres qui le composent.

Exemple :

$$\begin{aligned} 153 &= 1^3 + 5^3 + 3^3 \\ &= 1 + 125 + 27 \\ &= 153 \end{aligned}$$

153 est donc un nombre d'Armstrong. .

« 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. ⌚30'

1. Qu'affichent les deux programmes suivants :

```
1 /////////////// Programme 1 ///////////////
2 int main(){
3     int num1 = 1, num2 = 2;
4     switch(num1 > num2)
5     {
6         case 0:
7             printf("%d", num2); break;
8         case 1:
9             printf("%d", num1); break;
10        default:
11            printf("%d", num1+num2); break;
12    }
13 }
```

```
1 /////////////// Programme 2 ///////////////
2 #include <stdio.h>
3 int main()
4 {
5     int num = 123, toto = 0;
6     while(num != 0)
7     {
8         toto = (toto*10)+(num%10);
9         num /= 10;
10    }
11    printf("toto = %d", toto);
12    return 0;
13 }
```

2. Que font les deux programmes Programme 1 et Programme 2? (deux lignes au maximum pour chacun d'entre eux.)

Solution

1.

2 Affichage

toto = 321 Affichage

2. Le Programme 1 affiche le **maximum** entre les deux nombres num1 et num2. Le Programme 2 calcule puis affiche l'**inverse** d'un nombre entier.

2 Affichage de chiffres

☑ pts. ⌚30'

Écrire un programme qui demande à l'utilisateur un entier N positif puis affiche le motif suivant avec les chiffres allant de 1 à N..

Par exemple : N = 5

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

Solution

```

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

```

3 Nombre d'Armstrong

pts. 30'

Écrire une fonction qui détermine si un nombre entier positif donné comme paramètre est un nombre d'Armstrong, puis écrire un programme pour tester cette fonction.

On rappelle qu'un nombre d'Armstrong est un entier naturel qui est égal à la somme des cubes des chiffres qui le composent.

Exemple :

$$\begin{aligned}
 153 &= 1^3 + 5^3 + 3^3 \\
 &= 1 + 125 + 27 \\
 &= 153
 \end{aligned}$$

153 est donc un nombre d'Armstrong. .

Solution

```

1 #include <stdio.h>
2
3 int Est_Armstrong(int num)
4 {
5     int temp, dernierChiffre, somme;
6     temp = num;
7     somme = 0;
8     while(temp != 0)
9     {
10        dernierChiffre = temp % 10;
11        somme += dernierChiffre * dernierChiffre * dernierChiffre;
12        temp /= 10;
13    }
14    if(num == somme)
15        return 1;
16    else
17        return 0;
18 }

```

```
1 int main()
2 {
3     int n ;
4     printf("Donner un entier");
5     scanf("%d", &n);
6     if (Est_Armstrong(n))
7         printf("%d est un nombre d'Armstrong", n);
8     else
9         printf("%d n'est pas un nombre d'Armstrong", n);
10    return 0;
11 }
```