



Contrôle Continu

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'

Qu'affiche les deux programmes suivants :

```
1 void Toto(int X, int *Y, int *Z)
2 {   int T;
3     T = X;
4     X = *Y;
5     *Y = *Z;
6     *Z = T;
7 }
8 void main()
9 {   int A=2, B=4, C=6;
10    Toto(A, &B, &C);
11    printf("%d %d %d \n", A, B, C);
12    Toto(A, &B, &C);
13    printf("%d %d %d \n", A, B, C);
14 }
```

```
1 void main ()
2 {   char B[7]="Loulou";
3     int A[3][4] = { {1, 12, 23, 34},
4                   {45, 56, 67}};
5     char *p = strchr(B, 'l');
6     printf ("%d %d\n", A[2][2], **A);
7     printf ("%d\n", *((A+1)+1) );
8     printf ("%d\n", *((A+1)+1) );
9     printf ("%c\n", *p);
10    printf ("%s\n", p);
11    printf ("%d\n", p-B);
12    *p = '\0';
13    printf ("%s\n", B);
14 }
```

Solution



2 Suppression des occurrences d'un élément

☑️ pts. ⌚30'

Écrire une fonction **suppression** qui prend en entrée un tableau **T** d'entier à une dimension, sa taille **L** (qui doit être strictement inférieur à 100) et un entier **N**. Cette fonction permet de supprimer toutes les occurrences de l'élément **N** dans le tableau **T**.

Exemple : avec $L = 10$ après `suppression(T, L, 4)`, L devient égale à 7.

T:

3	4	11	17	26	4	48	4	61	9
---	---	----	----	----	---	----	---	----	---

 → T:

3	11	17	26	48	61	9
---	----	----	----	----	----	---

☞ Remarque.

1. On ne demande ni la saisie du tableau **T** ni son affichage.
2. À la fin, toutes les cases du tableau **T** doivent être contiguës.
3. La taille **L** du tableau **T** doit être mise à jour à la fin de la fonction.

Solution

Bon Courage !

```

1 void suppression(int T[100], int *L ,int N)
2 {
3     int i, j;
4     for(i=*L;i>=0;i--)
5     {
6         if(T[i]==N)
7         {
8             j=i;
9             while(j<*L)
10            {
11                T[j]=T[j+1];
12                j++;
13            }
14            (*L)--;
15        }
16    }
17 }

```

3 Triangle de Pascal

📄 pts. ⌚35'

Écrire une fonction qui construit le triangle de Pascal de degré N et le mémorise dans un tableau P de deux dimensions. La fonction doit également afficher le triangle de Pascal exactement comme indiqué dans l'exemple ci-dessous :

Exemple : N = 6 :

```

n=0  1
n=1  1  1
n=2  1  2  1
n=3  1  3  3  1
n=4  1  4  6  4  1
n=5  1  5  10 10 5  1
n=6  1  6  15 20 15 6  1

```

📖 Méthode :

- Calculer et afficher seulement les valeurs jusqu'à la diagonale principale (inclusive).
- Les valeurs de la première colonne et de la diagonale principale sont égales à 1.
- Les autres valeurs sont calculées de gauche à droite en utilisant la relation : $P_{i,j} = P_{i-1,j} + P_{i-1,j-1}$

Solution

```

1 void Triangle_Pascal(int P[14][14], int N)
2 {
3     int i, j;
4     for (i=0; i<=N; i++)
5     {
6         P[i][i]=1;
7         P[i][0]=1;
8         for (j=1; j<i; j++)
9             P[i][j] = P[i-1][j] + P[i-1][j-1];
10    }
11    printf("Triangle de Pascal de degre %d :\n", N);
12    for (i=0; i<=N; i++)
13    {
14        printf(" N=%2d", i);
15        for (j=0; j<=i; j++)
16            if (P[i][j])
17                printf("%5d", P[i][j]);
18        printf("\n");
19    }
20 }

```