

TP N° 4 : Boucles et tableaux unidimensionnels

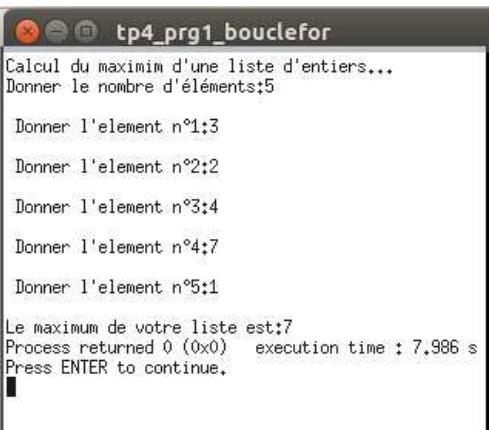
But du TP :

1. Utiliser les boucles ;
2. Utiliser les tableaux unidimensionnels

Énoncé :

Partie I: Boucles

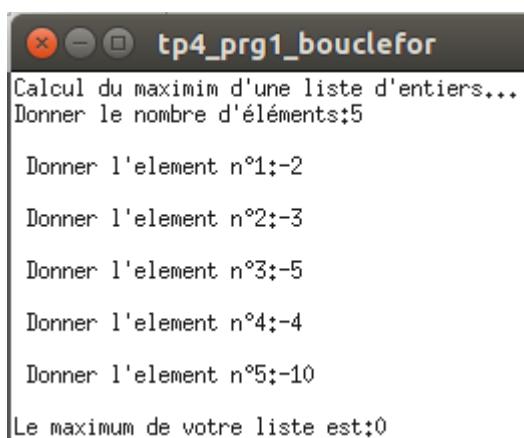
1. Exécuter le programme suivant avec la liste de 5 éléments : 3,2,4,7,1 puis l'exécuter une 2^{ème} fois avec les 5 entiers : -2,-3,-5,-4 et -10.



```

1 #include <stdio.h>
2 int n,max,i,nbr;
3 int main()
4 {
5     printf("Calcul du maximum d'une liste d'entiers...\n");
6     printf("Donner le nombre d'éléments:");
7     scanf("%d",&n);
8     max=0;
9     for(i=1;i<=n;i++){
10         printf("\n Donner l'element n°%d:",i);
11         scanf("%d",&nbr);
12         if(nbr>max){max=nbr;}
13     }
14     printf("\nLe maximum de votre liste est:%d", max);
15     return 0;
16 }
```

- Pourquoi le programme marche bien pour la 1iere liste mais donne une fausse réponse dans la deuxième ?



```

Calcul du maximum d'une liste d'entiers...
Donner le nombre d'éléments:5

Donner l'element n°1:-2
Donner l'element n°2:-3
Donner l'element n°3:-5
Donner l'element n°4:-4
Donner l'element n°5:-10

Le maximum de votre liste est:0
```

- Afin de corriger le programme pour qu'il affiche le maximum de la liste quel que soit le signe de ses éléments, il suffit d'initialiser max par le premier élément de la liste. Cette solution peut être mise en œuvre selon les 2 méthodes :
 - Méthode 1 : initialiser max par le premier élément à lire hors boucle.
 - Méthode 2 : initialiser max dans la boucle par l'élément lu lorsqu'il s'agit du 1^{ier}.
- Proposer un code pour chacune de ces méthodes.
- Modifier le programme pour qu'il affiche, en plus, le rang du maximum trouvé dans la liste.

```

Calcul du maximum d'une liste d'entiers...
Donner le nombre d'éléments:5

Donner l'element n°1:-2
Donner l'element n°2:-3
Donner l'element n°3:-5
Donner l'element n°4:-4
Donner l'element n°5:-10

Le maximum de votre liste est:-2 et son rang est 1

```

2. Exécuter le programme suivant :

```

1 #include <stdio.h>
2 #include <stdbool.h> //pour pouvoir déclarer le booléen
3
4 int chiffUnit,chiffDiz;
5 bool nonConform;
6
7 int main(){
8
9     do{
10         printf("\n Donner un nbr positif de deux chiffres SVP:");
11         scanf("%d",&chiffUnit);
12         if((nonConform==((chiffUnit>99)|||chiffUnit<10))){printf("\n Non conforme!");}
13     }while(nonConform);
14
15     chiffDiz=0;
16     while(chiffUnit>=10){
17         chiffUnit-=10;
18         chiffDiz++;
19     }
20
21     //for(chiffDiz=0;chiffUnit>=10;chiffUnit-=10,chiffDiz++)
22
23     printf("\n résultat %d",10*chiffUnit+chiffDiz);
24     return 0;
25 }

```

- Expliquer ce qu'il fait.
- Commenter les points suivants :
 - Le header stdbool.h.
 - Rôle de la boucle do while.
 - Accès inconditionnel à la boucle do while.
 - La balisage de cette boucle par les accolades est-il nécessaire ?
 - Rôle de la boucle while.
 - Résultat affiché par l'algorithme.

Partie II : Tableaux unidimensionnels

Soit le programme suivant :

- Qu'est ce qu'il fait ?
- Expliquer ce qui suit :
 - Comment déclarer des tableaux unidimensionnels.
 - Comment les initialiser. Sans cette initialisation quelles seraient les valeurs ? Si le nombre de valeurs d'initialisation dépasse la taille.
 - Comment accéder à un élément.
 - Quel est l'intervalle des indices ?
 - Pourquoi dans la ligne 17 on affiche i+1 ?

```
1 #include <stdio.h>
2 #define TailleMax 20
3 int n,max,i,vectNbr[TailleMax],imax;
4 int main(){
5     printf("Calcul du maximum d'une liste d'entiers...\n"
6           "Donner le nombre d'éléments(<%d):",TailleMax);
7     scanf("%d",&n);
8     if(n>TailleMax){
9         printf("\n %d est supérieur à %d! Arrêt d'exécution.",
10             n,TailleMax);
11        return 1; //enfin un return diff de zero :)
12    }
13    //lecture des éléments
14    printf("\n Donner l'élément n°1:");
15    scanf("%d",&vectNbr[0]);
16    for(i=1;i<n;i++){
17        printf("\n Donner l'élément n°%d:",i+1);
18        scanf("%d",&vectNbr[i]);
19    }
20    //recherche du max
21    max=vectNbr[0];imax=1;
22    for(i=1;i<n;i++){
23        if(vectNbr[i]>max){max=vectNbr[i];imax=i+1;}
24    }
25    | printf("\nLe maximum de votre liste est:%d et son rang est %d",max,imax);
26    return 0;
27 }
```

- Quelle est la différence avec le programme 1 (après modif demandée) de la Partie I ?
- Exécuter le programme avec une liste d'éléments (par exemple 3,2,4,7,1,-2,3,-5,14 et -10).

Partie III : Les notes possibles d'un examen sont les valeurs entières de 0 à 10. Pour faire des statistiques dans une école, on voudrait connaître pour chaque note combien d'élèves l'ont eue. En supposant avoir n élèves, écrire un programme permettant d'effectuer les opérations suivantes :

1. Saisir les n notes des élèves pour les mettre dans un tableau **tab**. La note sera rejetée tant qu'elle n'est pas possible.
2. Construire le tableau **stat** tel que **stat[j]** représente le nombre d'élèves qui ont eu la note **j**.
3. Afficher le tableau **stat** de la manière suivante :

Note	Nombre d'élèves
0	
1	*
2	***
3	*****
4	**
5	*****
6	*****
7	****
8	**
9	*
10	*

4. Afficher le nombre d'élèves ayant obtenu la plus bonne note.

Partie IV: Exercices à faire en C

Exercice 1 : Ecrire un programme qui détermine le nombre d'occurrences de chaque élément d'un tableau d'entiers.

Exercice 2 : Soient deux vecteurs vectA et vectB ayant le même nombre d'éléments. Ecrire un programme qui permet de construire deux vecteurs vMax et vMin définis par: vMax(i) = maximum entre vectA(i) et vectB(i) et vMin(i) = minimum entre vectA(i) et vectB(i). On comptabilisera au fur et à mesure les cas d'égalités.

Exercice 3 : Soit vect un tableau de nombres entiers triés dans l'ordre croissant. Écrire un programme qui insère un nombre nbr dans ce tableau, tout en conservant le tri.