

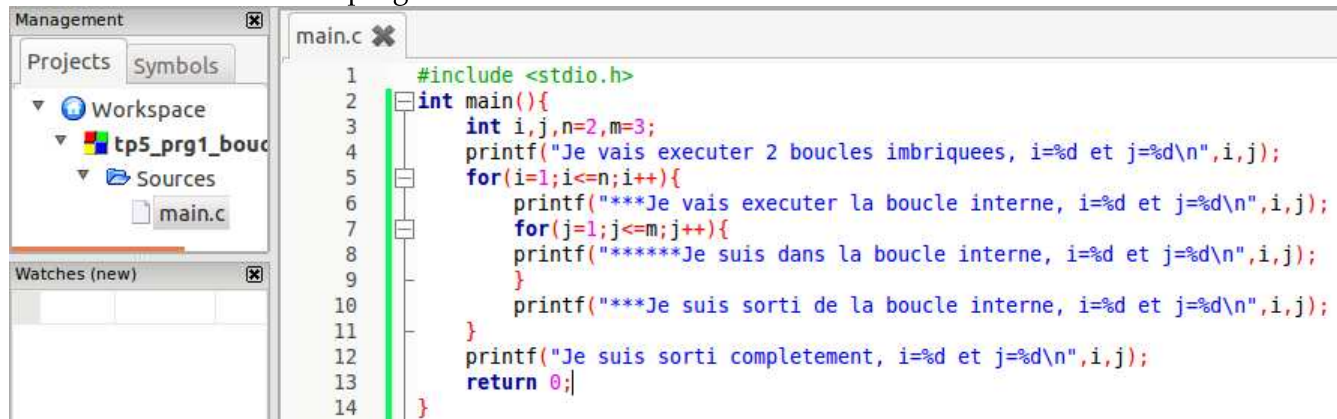
## TP N° 5 : Boucles imbriquées et tableaux bidimensionnels

### But du TP :

1. Utiliser les boucles imbriquées
2. Utiliser les tableaux bidimensionnels.

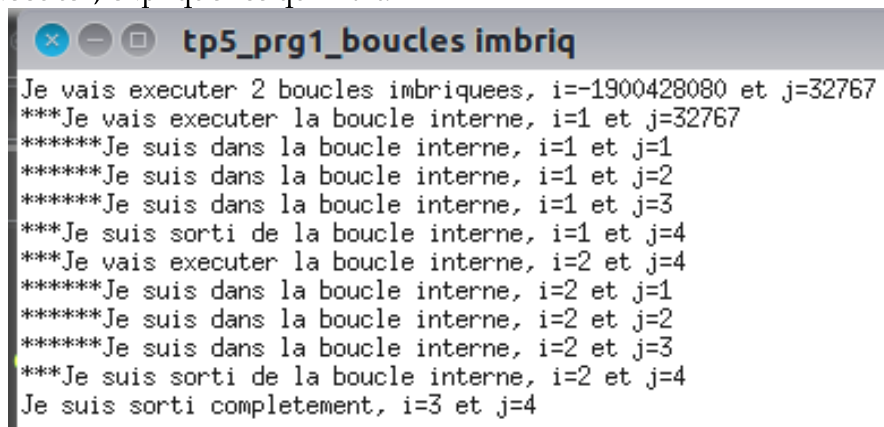
### Énoncé :

#### Partie I: Considérons le programme suivant :



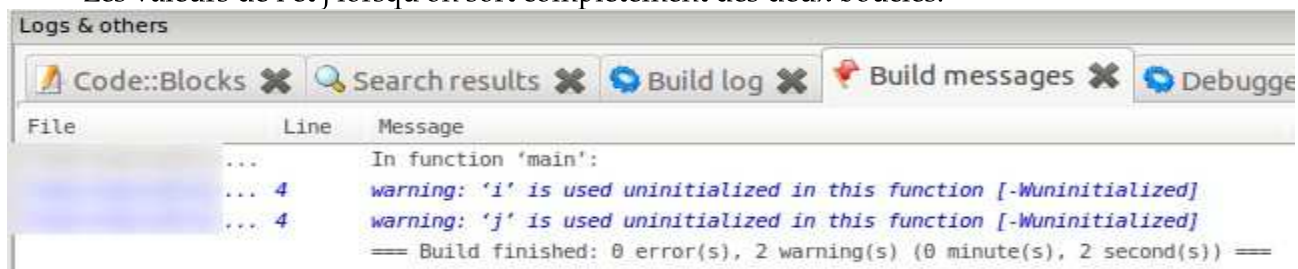
```
1  #include <stdio.h>
2  int main(){
3      int i,j,n=2,m=3;
4      printf("Je vais executer 2 boucles imbriquees, i=%d et j=%d\n",i,j);
5      for(i=1;i<=n;i++){
6          printf("***Je vais executer la boucle interne, i=%d et j=%d\n",i,j);
7          for(j=1;j<=m;j++){
8              printf("*****Je suis dans la boucle interne, i=%d et j=%d\n",i,j);
9          }
10         printf("***Je suis sorti de la boucle interne, i=%d et j=%d\n",i,j);
11     }
12     printf("Je suis sorti completement, i=%d et j=%d\n",i,j);
13     return 0;
14 }
```

1. Avant de l'exécuter, expliquer ce qu'il fait.



```
Je vais executer 2 boucles imbriquees, i=-1900428080 et j=32767
***Je vais executer la boucle interne, i=1 et j=32767
*****Je suis dans la boucle interne, i=1 et j=1
*****Je suis dans la boucle interne, i=1 et j=2
*****Je suis dans la boucle interne, i=1 et j=3
***Je suis sorti de la boucle interne, i=1 et j=4
***Je vais executer la boucle interne, i=2 et j=4
*****Je suis dans la boucle interne, i=2 et j=1
*****Je suis dans la boucle interne, i=2 et j=2
*****Je suis dans la boucle interne, i=2 et j=3
***Je suis sorti de la boucle interne, i=2 et j=4
Je suis sorti completement, i=3 et j=4
```

2. Exécuter le programme pour différentes valeurs de n et m puis commenter les points suivants :
  - Combien de fois la boucle interne est exécutée.
  - Les valeurs de i et j ne changent pas simultanément.
  - Les valeurs de i et j avant d'entrer aux boucles.
  - La valeur de j chaque fois qu'on sort de la boucle interne.
  - Les valeurs de i et j lorsqu'on sort complètement des deux boucles.



File	Line	Message
...	...	In function 'main':
...	4	warning: 'i' is used uninitialized in this function [-Wuninitialized]
...	4	warning: 'j' is used uninitialized in this function [-Wuninitialized]
== Build finished: 0 error(s), 2 warning(s) (0 minute(s), 2 second(s)) ==		

3. Lors de la compilation, il y avait deux warnings (mise en garde). Identifiez-les dans le log de compilation. Qu'est-ce qu'ils signifient ?
4. Commenter les lignes 4 et 6 (mettre // au début de chaque ligne), recompiler. Est-ce que les warnings disparaissent ?

**Partie II :** Considérons le programme suivant :

The screenshot shows a C program in a text editor and its execution output in a terminal window.

```

1  #include <stdio.h>
2  int main(){
3      int i,j,n=3;
4      printf("Je vais executer une boucle, i=%d et j=%d\n",i,j);
5      j=1;
6      for(i=1;i<=n;i++){
7          //j=1;
8          printf("***Je suis dans la boucle, i=%d et j=%d\n",i,j);
9          j++;
10     }
11     printf("Je suis sorti, i=%d et j=%d\n",i,j);
12     return 0;
13 }
  
```

The terminal output shows the program's execution:

```

tp5_prg2_boucle2indice
Je vais executer une boucle, i=32765 et j=0
***Je suis dans la boucle, i=1 et j=1
***Je suis dans la boucle, i=2 et j=2
***Je suis dans la boucle, i=3 et j=3
Je suis sorti, i=4 et j=4
Process returned 0 (0x0)   execution time : 0,006 s
Press ENTER to continue.
  
```

1. Avant de l'exécuter, expliquer ce qu'il fait.
2. Exécuter le programme pour différentes valeurs de n puis commenter les points suivants :
  - Les valeurs de i et j changent simultanément.
  - Les valeurs de i et j avant d'entrer dans la boucle.
  - Les valeurs de i et j lorsqu'on sort de la boucle.
  - Enlever les // de la ligne 7. Exécuter puis commenter l'affichage.

### Partie III: Introduction aux tableaux bidimensionnels

Un tableau bidimensionnel (qu'on appellera dans ce qui suit matrice) est une structure de données qui permet de représenter des données de même type sous forme tabulaire. On utilisera deux indices pour accéder à un élément de cette matrice. Le tableau suivant représente les valeurs des indices [i,j] d'une matrice 5x6 (i.e. 5 lignes et 6 colonnes).

1. Compléter ce tableau

[0,0]	[0,1]				
[2,0]					

2. Considérons le programme 1 de la première partie pour n=5 et m=6. On suppose que nous allons accéder à mat[i][j] dans la boucle interne. Utiliser le tableau précédent pour identifier le parcours réalisé dans la matrice.
3. Considérons le programme 1 de la deuxième partie pour n=5 et m=6. On suppose que nous allons accéder à mat[i][j] dans la boucle. Utiliser le tableau précédent pour identifier le parcours réalisé dans la matrice.
4. Le code suivant remplit une matrice ligne par ligne par les valeurs entières de 1 à nxm. Identifier dans ce code :

```

1  #include <stdio.h>
2  int main(){
3      int i,j,n=5,m=6,val,mat[10][10];
4      //remplissage
5      val=1;
6      for(i=0;i<n;i++){
7          printf("***Je vais passer à la ligne i=%d\n",i);
8          for(j=0;j<m;j++){
9              mat[i][j]=val++;
10             printf("*****Je suis dans la boucle interne, j'ai mis"
11                 " la valeur %2d dans la case [%d][%d]\n",mat[i][j],i,j);
12         }
13         printf("***J'ai terminé le parcours de la ligne %d\n\n",i);
14     }
15     printf("Je suis sorti complètement\n\n\n La matrice remplie est :\n");
16     //Affichage
17     for(i=0;i<n;i++){
18         for(j=0;j<m;j++){
19             printf("%3d",mat[i][j]);
20         }
21         printf("\n");
22     }
23     return 0;
24 }

```

- comment on déclare une matrice.
- comment on accède à ses éléments.
- les valeurs des indices associées à la première ligne, 1ière colonne, dernière ligne et dernière colonnes.
- le résultat de l'affectation de la ligne 9 (essayer avec ++val pour voir la différence).
- le moyen qui a permis d'afficher la matrice de manière tabulée.

5. Modifier le code pour qu'il réalise le remplissage colonne par colonne.

#### Partie IV :

Écrire un programme C pour chacun des problèmes suivants :

- compter dans une matrice le nombre de valeurs positives ou nulles et remplacer toute valeur négative par sa valeur absolue.
- calculer la somme des éléments de la première diagonale d'une matrice carrée.
- calculer la somme des éléments de la deuxième diagonale d'une matrice carrée.
- permuter les éléments de deux lignes données d'une matrice.