

Contraintes

Les solutions au DS sont validées via la plateforme domjudge. Chaque élève a son propre login et son propre mot de passe. Vous pouvez envoyer plusieurs fois une solution pour un même problème sans aucune pénalité.

Vous avez le droit aux supports des cours, TD, TP et aussi à des ressources Internet, mais vous n'avez pas le droit de communiquer avec d'autres personnes. La DSI sauvegarde tout le trafic TCP/IP pendant le DS. Vous pouvez utiliser seulement le protocole http pour accéder aux différents sites Web. Une connexion à un outil de messagerie, réseau social etc invalide automatiquement votre participation au DS avec les conséquences prévues par le règlement des études.

Vous pouvez utiliser le langage C ou le C++, mais sans utiliser la STL.

Problème 3 : nombre de chemins pour un robot - 6 points

Un robot ne peut se déplacer que vers la droite ou vers le bas. Il part du coin en haut à gauche et doit aller au coin en bas à droite. Le plateau sur lequel il évolue est indiqué par une grille avec des passages possibles (caractère '+') et des obstacles (caractère 'O', lettre o majuscule). Vous devez trouver le nombre de chemins possibles pour le robot. Par convention, si le plateau contient un obstacle au départ ou à l'arrivée le nombre renvoyé sera 0.

En entrée :

- un entier n indiquant la taille du plateau
- le plateau sous la forme de n lignes de n caractères '+' ou 'O'

En sortie :

Le nombre de chemins possibles suivi de "\r\n"

* Exemple 1

Entrée

7

```
+++++++  
+OOOOO+  
+OOOOO+  
+OOOOO+  
+OOOOO+  
+OOOOO+  
+OOOOO+  
+++++++
```

Sortie :

2

* Exemple 2

Entrée :

3

```
+++  
+++  
+++
```

Sortie :
6

* Exemple 3

Entrée :

10
+++++00000
+++++00000
+++++00000
+++++00000
+++++00000
00000+++++
00000+++++
00000+++++
00000+++++
00000+++++

Sortie :
4900

Entrées-sorties

On pourra utiliser l'ossature suivante pour lire le tableau en entrée et afficher le résultat.
Dans ce code, on accède à la case i,j grâce à tab[i][j] qui est un char qui vaut '+' ou 'O' :

```
#include <stdio.h>
#include <stdlib.h>

char ** read_tab(int n) {
    int i,j;
    char ** tab;
    tab = (char **) malloc(sizeof(char*)*n);
    for (i=0;i<n;i++) {
        tab[i] = (char*) malloc(sizeof(char)*n);
        for (j=0;j<n;j++) {
            tab[i][j] = getchar();
        }
        getchar();
    }
    return tab;
}

int main(void) {
    int n;
    char ** tab;
    int npossibilites = 0;
    scanf("%d",&n);
    getchar();
    tab = read_tab(n);
    /* compute npossibilites here! */
    printf("%d\\r\\n",npossibilites);
    return 0;
}
```

Observation

Ne pas oublier la ligne :

return 0;

dans votre fonction main !