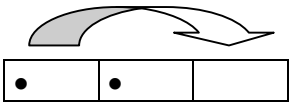
## Projet : Le solitaire

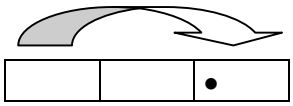
On considère un tableau d'entiers à deux dimensions représentant un jeu de solitaire en croix : les cases vides sons représentées par des zéros, les cases occupées par des pions par la valeur 1 et les cases inexistantes par la valeur ï 1. La figure ci-dessous représente ce tableau dans sa position de départ : toutes les cases sont occupées sauf la case centrale ; la partie du tableau correspondant à la grille de jeu est encadrée de trais épais :

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| -1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 |
| -1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 |
| -1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| -1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 |
| -1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 |
| -1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 |

Le but du jeu est d'éliminer un à un les pions pour n'en avoir plus qu'un seul à la fin de la partie. Chaque coup consiste à supprimer un pion ; pour le faire on fait "sauter" un pion pardessus son voisin direct (horizontalement ou verticalement) ; pour cela la case voisine opposée au pion "sauté" doit être vide.



avant



après

Le pion de droite s'est déplacé à gauche et le pion du milieu a été supprimé

Après N coups, la grille de jeu comporte donc N+1 cases vides. La partie se termine lorsqu'on ne peut plus éliminer de pion (on a alors perdu), c'est-à-dire qu'aucun pion n'a de voisin direct vertical ou horizontal suivi d'une case libre ou s'il ne reste plus qu'un seul pion (on a alors gagné).

Soit G le tableau représentant le jeu :

```
const int NBCASES = 9 ;

array<array<int,NBCASES>,NBCASES> Grille ;
```

1) Ecrire une fonction **initGrille** à un paramètre **g** représentant la grille qui initialise le tableau g avec des valeurs ï 1.

2) Ecrire une fonction **remplirGrille** à un paramètre **g** représentant la type Grille qui remplit le solitaire avec les valeurs 1 et 0 (case centrale pour le 0).

3) Ecrire une fonction **afficherGrille** à un paramètre g représentant la grille qui affiche le solitaire sous la forme suivante :

```
        1 1 1
        1 1 1
        1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 1 1
        1 1 1
        1 1 1
        1 1 1
```

4) Ecrire une fonction **nbPions** à un paramètre g représentant la grille qui retourne le nombre de pions présents sur la grille g.

   Exemple : pour la grille de départ , NbPions(g) → 44

5) Ecrire une fonction **estPriseHD** à 3 paramètres G de type Grille et x, y de type entier naturel qui retourne vrai si le pion (x, y) peut prendre horizontalement (h) et vers la droite (d) faux sinon

   Exemple: estPriseHD(G,5,3) → True
   estPriseHD(G,5,1) → False

6) Ecrire de même les fonctions **estPriseHG, estPriseVH** et **estPriseVB** respectivement Horizontalement vers la Gauche, Verticalement vers le Haut et Verticalement vers le Bas

7) Ecrire une fonction **estPrise** à 5 paramètres g représentant la grille, x, y de type entier et d, s de type caractère qui retourne true si le pion (x, y) peut prendre un pion selon le déplacement dep et le sens s sinon false

8) Ecrire une fonction **priseHD** à 3 paramètres g représentant la grille et x, y de type entier qui met à jour la grille g pour une prise par le pion (x, y) qui s'est faite horizontalement vers la droite

9) Ecrire de même les fonctions **PriseHG, PriseVH** et **PriseVB** respectivement pour Horizontalement Gauche, Verticalement Haut et Verticalement Bas

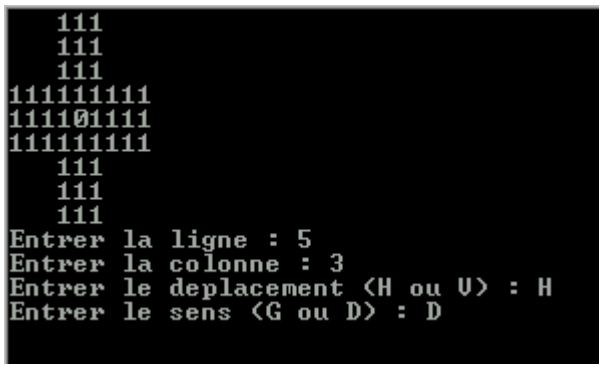
10) Ecrire une fonction **nbCoups** à 3 paramètres g représentant la grille et x, y de type entier qui retourne le nombre de coups possibles à partir d'une case (x, y) de la grille c'est-à-dire le nombre de pions différents pouvant être pris à partir de cette case.

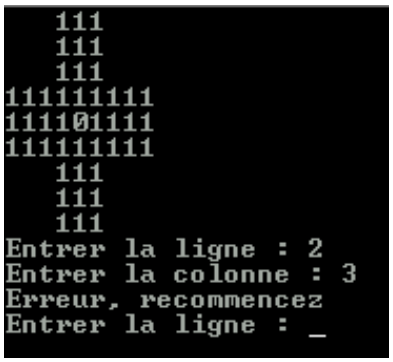
11) Ecrire une fonction **nbCoupsTotal** à un paramètre g représentant la grille qui désigne le nombre de coups possibles sur une grille donnée c'est-à-dire le nombre de possibilités différentes de prendre un pion. On pourra utiliser la fonction précédente

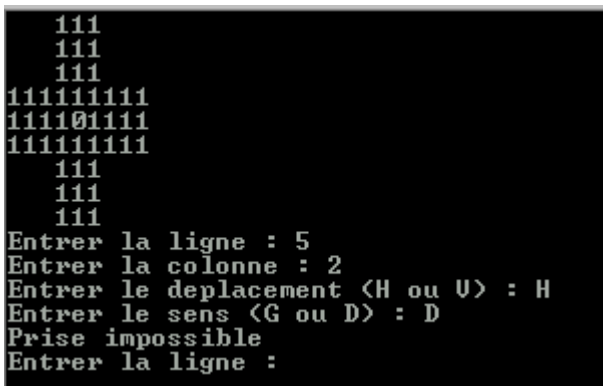
    Exemple : pour la grille de départ, nbCoupsTotal(g) → 4 possibilités car on ne peut éliminer que l'un des pions voisins de la case centrale

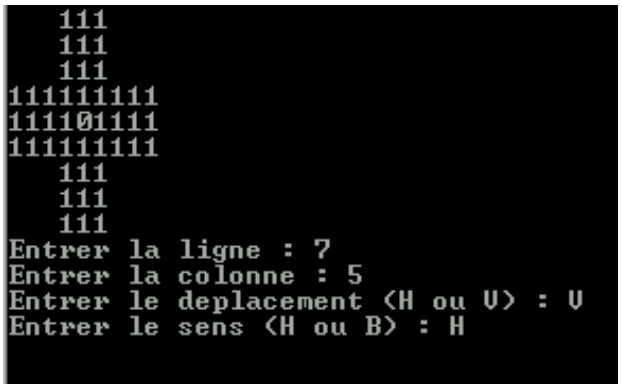
12) Ecrire une fonction **saisir** à 5 paramètres x, y de type entier, dep et sens de type caractère et g représentant la grille. Cette fonction demande à l'utilisateur un n° de ligne, un n° de colonne, le déplacement (h ou v) et le sens (d ou g pour horizontal et h ou b pour vertical). Les saisies seront contrôlées afin de bien sélectionner une case du solitaire. On recommence la saisie si erreur (par exemple la case (x, y) ne contient pas de pion) ou si le pion qui est sur la case (x, y) ne peut pas prendre. On pourra utiliser la fonction estPrise

Exemples :




13) Ecrire une fonction **JouerCoup** à 4 paramètres g représentant la grille, x et y de type entier, dep et sens de type caractère qui selon le déplacement (dep) et le sens, effectue la mise à jour de la grille g

14) Ecrire le programme principal permettant de jouer au solitaire. Un message "Perdu" ou "Gagne" doit apparaître en fin de partie. Dans le cas où la partie est perdue, on affichera le nombre de pions restant dans le jeu.