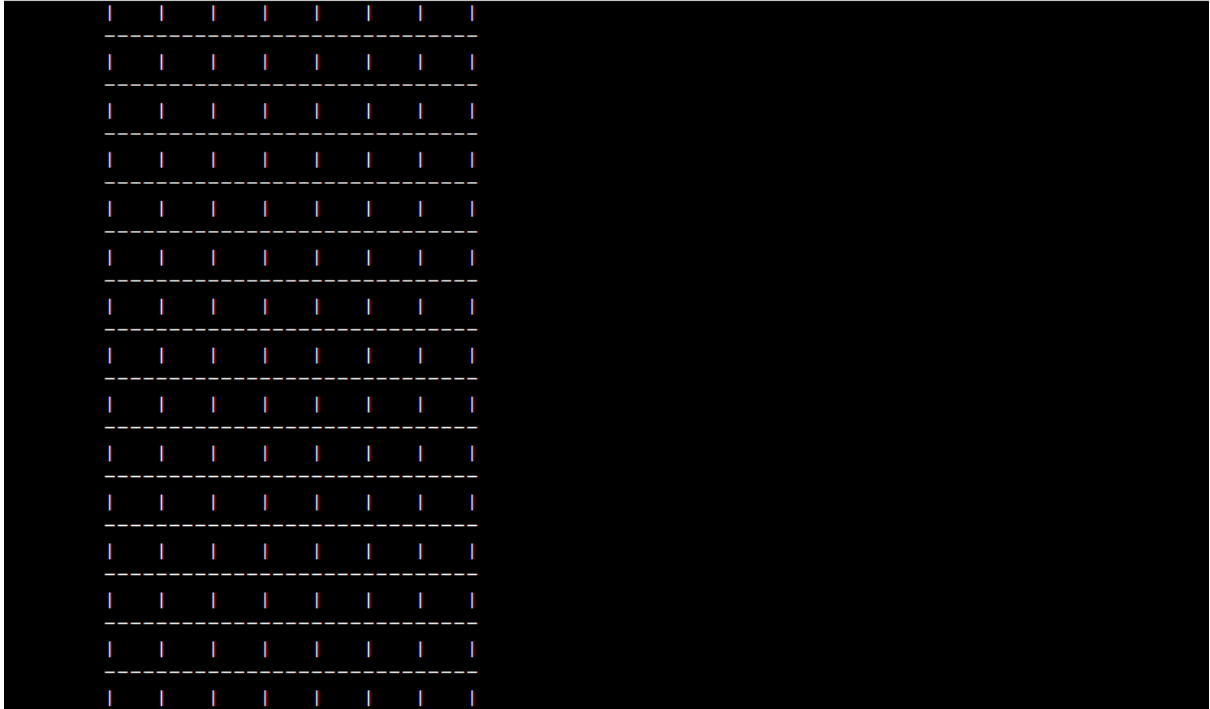


Cahier de test pour chaque résultat :

Procédure initGrille :

Cas 1 :

NBLIG = VIDE



Résultat obtenu :

affichage infini du tableau des lignes

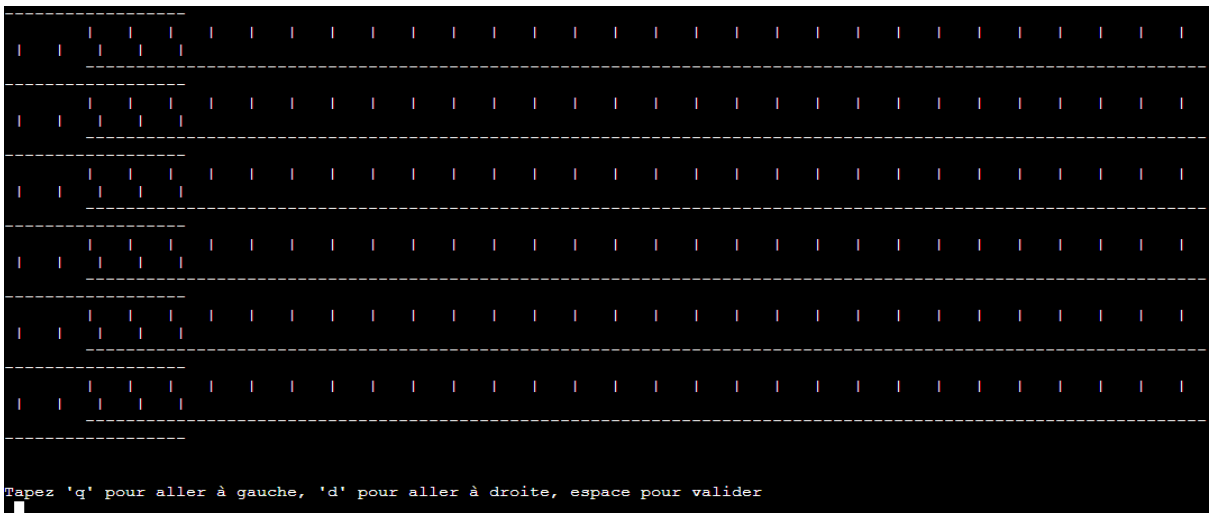
Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même.



Cas 2 :

NBCOL = VIDE



Résultat obtenu :

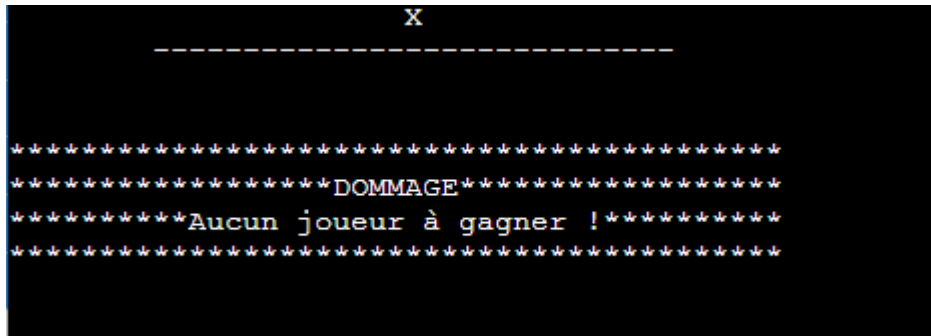
affichage infini du tableau des colonnes

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même.



Cas 3 : NBLIG = 0



Résultat obtenu :

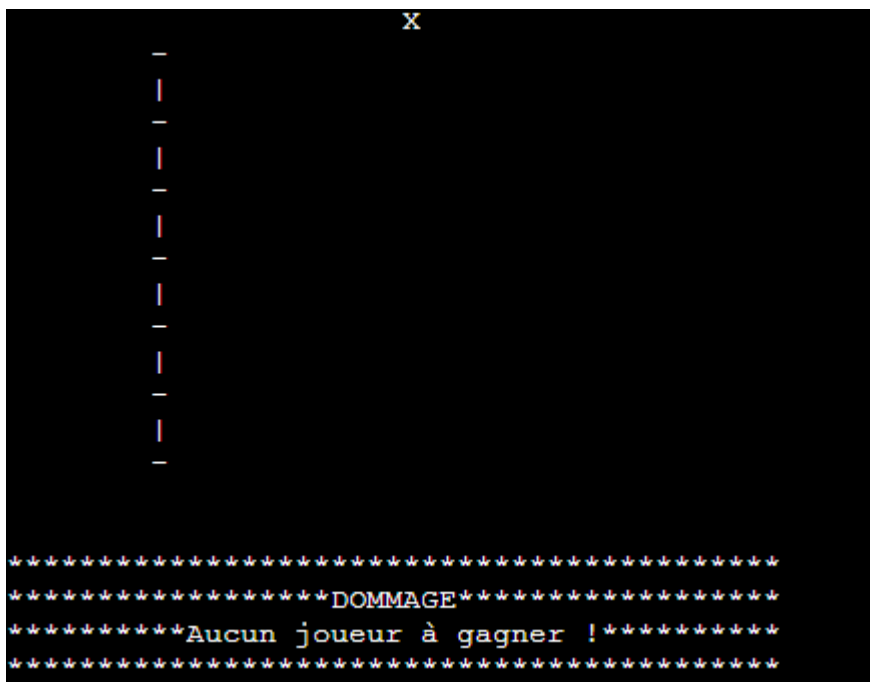
Il n'y a aucun affichage du tableau puisque qu'il n'y a pas de lignes et ça fait directement match nul

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même.



Cas 4 : NBCOL = 0



Résultat obtenu :

Il y a l'affichage que de la colonne 0

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même.

Cas 5 : NBCOL = a

```
main.c:6:19: error: 'a' undeclared here (not in a function)
  6 | #define NB_LIGNES a
    |                   ^
main.c:14:20: note: in expansion of macro 'NB_LIGNES'
 14 | typedef int Grille[NB_LIGNES][NB_COLONNES];
    |                   ^~~~~~
main.c:17:24: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 17 | void initGrille(Grille laGrille);
    |                   ^~~~~~
main.c:18:22: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 18 | void afficher(Grille laGrille, char pion);
    |                   ^~~~~~
main.c:19:26: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 19 | bool grillePleine(Grille laGrille);
    |                   ^~~~~~
main.c:20:19: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 20 | void jouer(Grille laGrille, char pion, int * ligne, int * colonne);
    |                   ^~~~~~
main.c:21:27: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 21 | int choisirColonne(Grille laGrille, char pion);
    |                   ^~~~~~
main.c:22:26: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 22 | int chercherLigne(Grille laGrille, int col);
    |                   ^~~~~~
```

Résultat obtenu :

Affiche plein d'erreur car le caractère ne peut pas servir de définition de tableau pour la ligne

Résultat du test :

Le test est réussi car le résultat attendu est le même 


Cas 6 : NBCOL = a

```
main.c:7:21: error: 'a' undeclared here (not in a function)
  7 | #define NB_COLONNES a
    |                   ^
main.c:14:31: note: in expansion of macro 'NB_COLONNES'
 14 | typedef int Grille[NB_LIGNES][NB_COLONNES];
    |                               ^~~~~~
main.c:17:24: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 17 | void initGrille(Grille laGrille);
    |                   ^~~~~~
main.c:18:22: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 18 | void afficher(Grille laGrille, char pion);
    |                   ^~~~~~
main.c:19:26: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 19 | bool grillePleine(Grille laGrille);
    |                   ^~~~~~
main.c:20:19: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 20 | void jouer(Grille laGrille, char pion, int * ligne, int * colonne);
    |                   ^~~~~~
main.c:21:27: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 21 | int choisirColonne(Grille laGrille, char pion);
    |                   ^~~~~~
main.c:22:26: warning: type defaults to 'int' in declaration of 'laGrille' [-Wimplicit-int]
 22 | int chercherLigne(Grille laGrille, int col);
    |                   ^~~~~~
```

Résultat obtenu :

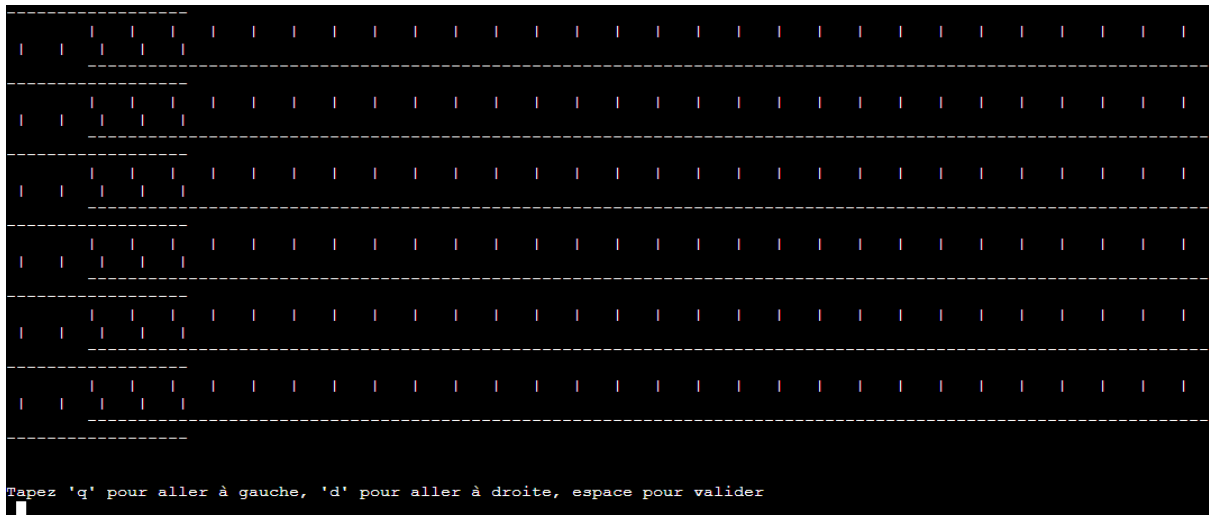
Affiche plein d'erreur car le caractère ne peut pas servir de définition de tableau pour la colonne

Résultat du test :

Le test est réussi car le résultat attendu est le même 

Procédure afficher


Cas 1 : Tableau = VIDE



Résultat obtenu :

affichage infini du tableau des lignes et des colonnes

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. 

Cas 2 : Tableau = de 0 à -1

```
main.c:14:13: error: size of array 'Grille' is negative
14 | typedef int Grille[NB_LIGNES][NB_COLONNES];
    |                  ^~~~~~
```

Résultat obtenu :

Erreur, il y'a une valeur négative dans le tableau donc

Résultat du test :

Le test est réussi car le résultat attendu est le même 

Cas 3 : Tableau = avec les bonnes valeurs de ligne et de colonne

```

                                x
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----

Tapez 'q' pour aller à gauche, 'd' pour aller à droite, espace pour valider
█
```

Résultat obtenu :

Il affiche bien le tableau comme il faut

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Cas 4 : COLONNE_DEBUT avec une valeur changée = pion à un autre emplacement (ex avec COLONNE_DEBUT = 4)

```

                                x
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----

Tapez 'q' pour aller à gauche, 'd' pour aller à droite, espace pour valider
█
```

Résultat obtenu :

Il affiche bien le pion à l'endroit demandé

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Cas 5 : COLONNE_DEBUT = 9

```

-----
| | | | | | | | |
-----
| | | | | | | | |
-----
| | | | | | | | |
-----
| | | | | | | | |
-----
| | | | | | | | |
-----
| | | | | | | | |
-----
*** stack smashing detected ***: terminated

```

Résultat obtenu :

stack smashing detected = erreur dépassement d'un tableau donc la valeur COLONNE_DEBUT = 9 est trop grande pour le tableau au dessus de la grille

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Cas 6 : COLONNE_DEBUT = a

```

main.c:34:27: error: 'a' undeclared here (not in a function)
 34 | const int COLONNE_DEBUT = a;
    |                           ^

```

Résultat obtenu :

Erreur car la constance COLONNE_DEBUT ne peut pas avoir a comme valeur car il doit obligatoirement avoir un nombre réel

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Cas 7 : PION_A = 1

```

      o
-----
| | | | | | | | |
-----
| | | | | | | | |
-----
| | | | | | | | |
-----
| | | | | | | | |
-----
| | | | | | | | |
-----
| | | | | | | | |
-----
Tapez 'q' pour aller à gauche, 'd' pour aller à droite, espace pour valider

```

Résultat obtenu :

Il affiche bien le pion A avec un caractère, mais comme 1 n'est pas un caractère alors il affiche rien (on peut le voir sur une case du bas qui est plus petite que les autres)

Résultat du test :

Le test n'est pas réussi car le résultat attendu n'est pas le même ✗

Cas 8 : PION_B = g

```

      g
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | g | | | |
-----
| | | x | x | | |
-----
Tapez 'q' pour aller à gauche, 'd' pour aller à droite, espace pour valider

```

Résultat obtenu :

Le pion B n'est plus 'O' mais 'g'

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Procédure grillePleine :

Procédure afficher

Cas 1 : Tableau = VIDE

```

| | | | | | | | | | | | | | | | | | | | | |
-----
| | | | | | | | | | | | | | | | | | | | | |
-----
| | | | | | | | | | | | | | | | | | | | | |
-----
| | | | | | | | | | | | | | | | | | | | | |
-----
| | | | | | | | | | | | | | | | | | | | | |
-----
| | | | | | | | | | | | | | | | | | | | | |
-----
Tapez 'q' pour aller à gauche, 'd' pour aller à droite, espace pour valider

```

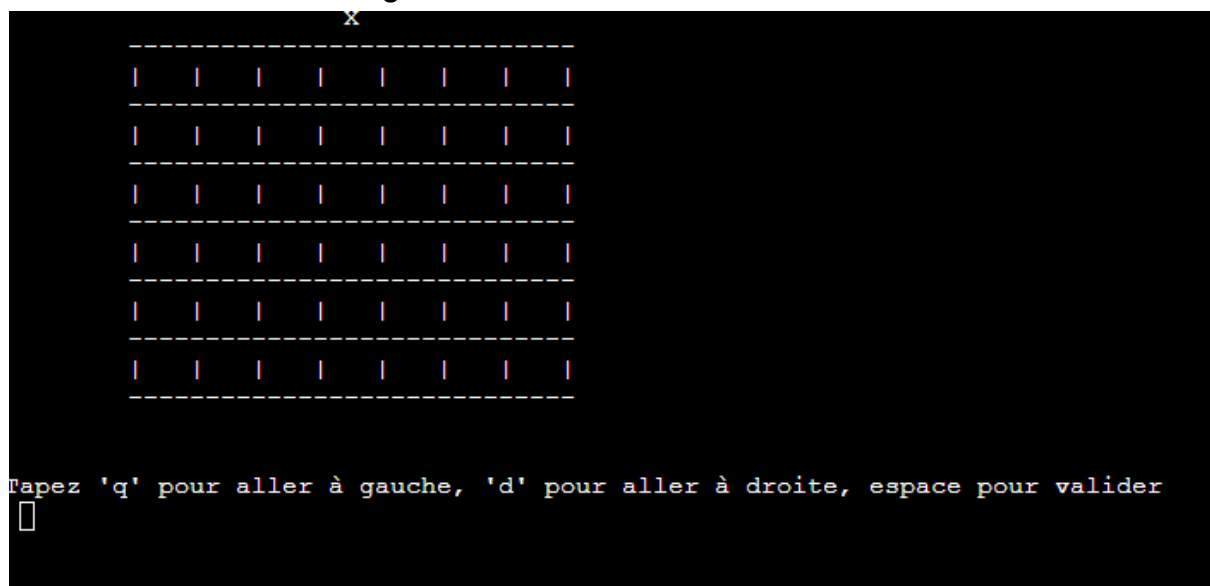
Résultat obtenu :

affichage infini du tableau des lignes et des colonnes

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. ✗

Cas 2 : Tableau = de 6 lignes et 7 colonnes



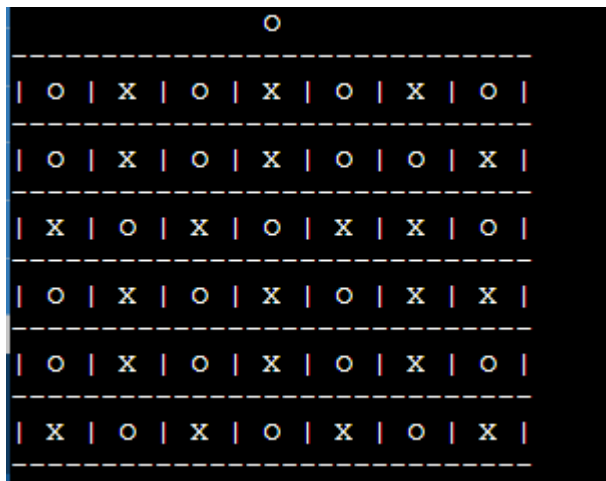
Résultat obtenu :

Il regarde si tout les case sont pleine ou non (ici non et alors renvoie faux la grille n'est pas pleine)

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Cas 3 : grille qui est pleine



Résultat obtenu :

Il n'a pas réussi à détecter que ma grille est pleine

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. ✗

Cas 4 : grille qui est pas pleine

```
x
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----

Tapez 'q' pour aller à gauche, 'd' pour aller à droite, espace pour valider
█
```

Résultat obtenu :

Il regarde si tout les case sont pleine ou non (ici non et alors renvoie faux la grille n'est pas pleine)

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Cas 5 : la valeur booléenne renvoie autre chose que du booléen

```
bool grillePleine(t_tableau Grille)
{
    int col, lgn;
    col=0;
    lgn=0;
    bool trouve;
    while((trouve!=true) || (col!=NBCOL))
    {
        if(Grille[lgn][col]==VIDE)
        {
            trouve = 2;
        }else
        {
            trouve = 5;
        }
        col++;
        lgn++;
    }
    return trouve;
}
```

```
o
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | x | | | | |
-----
| x | x | x | o | o | o | o |
-----
```

Résultat obtenu :

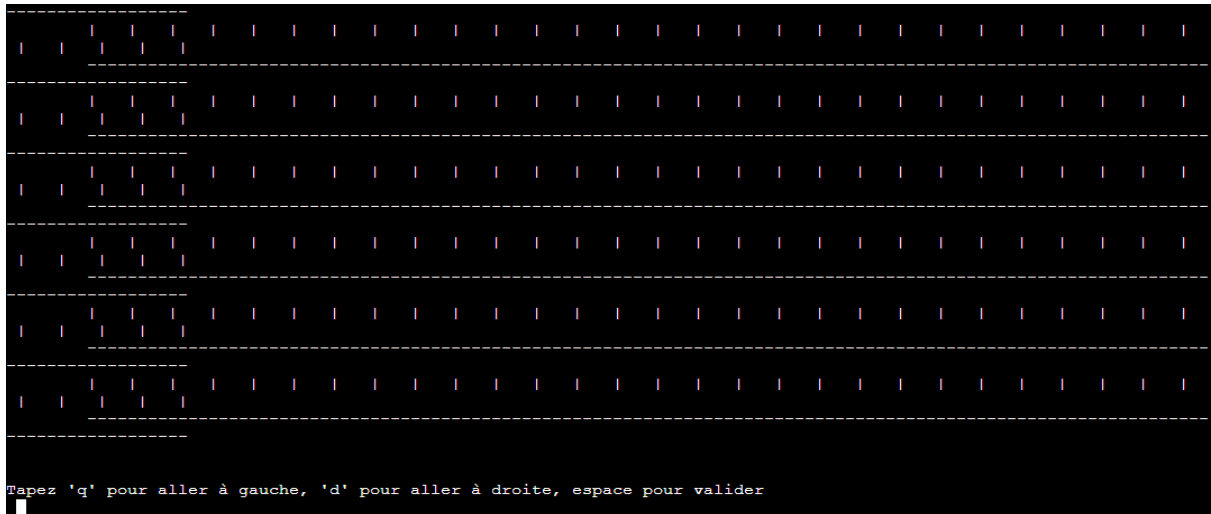
Il ne rentre je crois jamais dans le while donc ça marche pas

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. ✗

Procédure Jouer :

Cas 1 : Tableau = VIDE



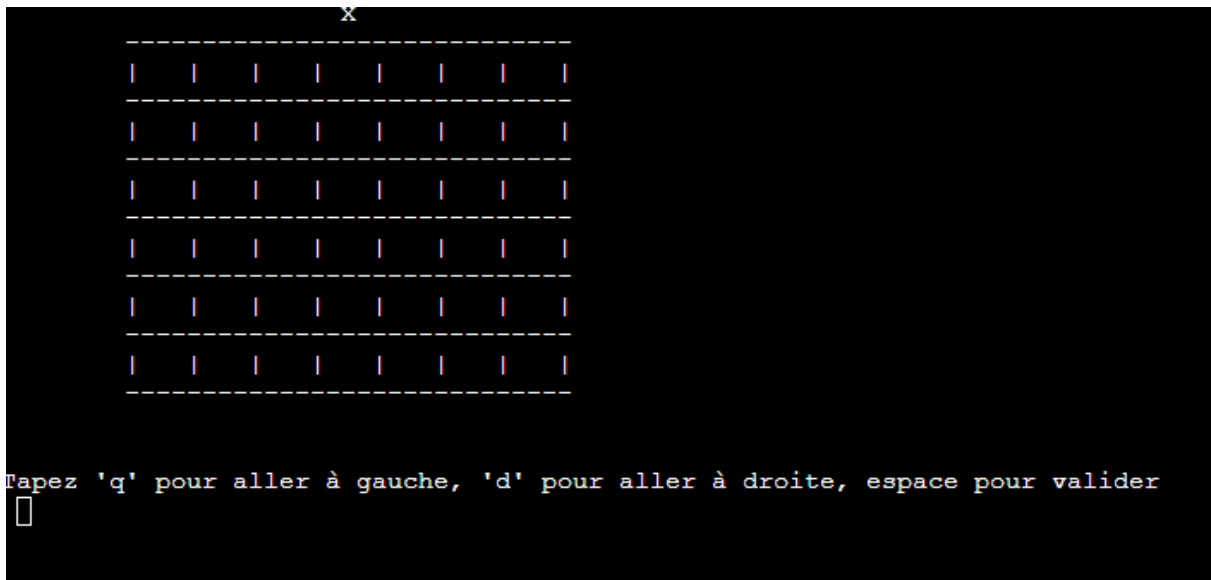
Résultat obtenu :

affichage infini du tableau des lignes et des colonnes

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. ❌

Cas 2 : Tableau = de 6 lignes et 7 colonnes



Résultat obtenu :

Il regarde si tout les case sont pleine ou non (ici non et alors renvoie faux la grille n'est pas pleine)

Résultat du test :

Le test est réussi car le résultat attendu est le même ✅

Cas 3 : PION_A = 1

```

      O
-----
|  |  |  |  |  |  |  |  |
-----
|  |  |  |  |  |  |  |  |
-----
|  |  |  |  |  |  |  |  |
-----
|  |  |  |  |  |  |  |  |
-----
|  |  |  |  |  |  |  |  |
-----
Tapez 'q' pour aller à gauche, 'd' pour aller à droite, espace pour valider

```

Résultat obtenu :

Il affiche bien le pion A avec un caractère, mais comme 1 n'est pas un caractère alors il affiche rien (on peut le voir sur une case du bas qui est plus petite que les autres)

Résultat du test :

Le test n'est pas réussi car le résultat attendu n'est pas le même ❌

Cas 4 : ligne = a

```

main.c: In function 'jouer':
main.c:194:17: error: 'a' undeclared (first use in this function)
  194 |         *_ligne=a;
      |         ^
main.c:194:17: note: each undeclared identifier is reported only once for each function it appears in

```

Résultat obtenu :

Erreur car ligne doit être égal à 0 ou 6

Résultat du test :

Le test est réussi car le résultat attendu est le même ✅

Cas 5 : colonne = b

```

main.c: In function 'jouer':
main.c:193:19: error: 'b' undeclared (first use in this function)
  193 |         *_colonne=b;
      |         ^
main.c:193:19: note: each undeclared identifier is reported only once for each function it appears in

```

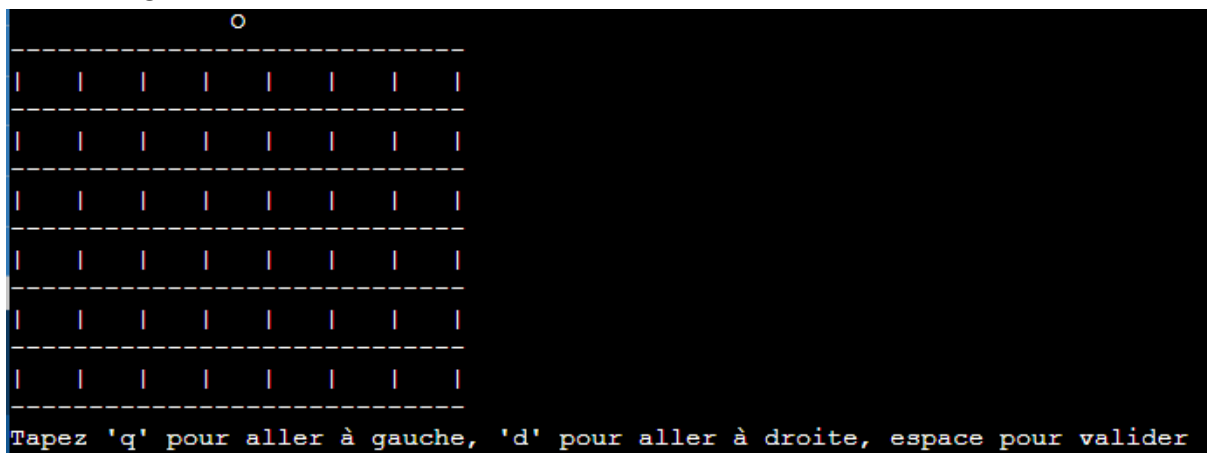
Résultat obtenu :

Erreur car colonne doit être égal à 0 ou 6

Résultat du test :

Le test est réussi car le résultat attendu est le même ✅

Cas 6 : ligne = 7



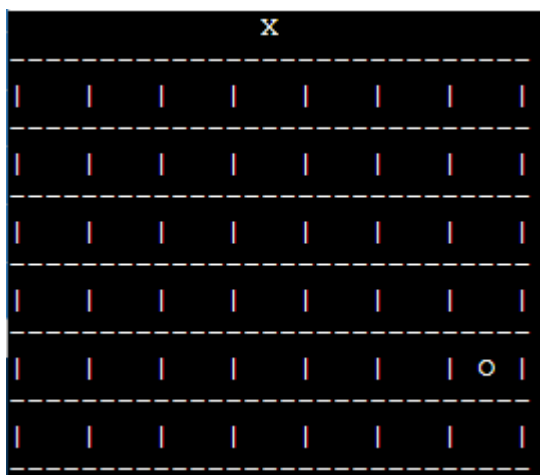
Résultat obtenu :

Il met le pion dans une valeur du tableau inconnu, il met le pion quelque part

Résultat du test :

Le test n'est pas réussi car le résultat attendu n'est pas le même ❌

Cas 7 : colonne = -1



Résultat obtenu :

Il met le pion dans la case du tableau où se situe le 'O' et toujours cette case

Résultat du test :

Le test n'est pas réussi car le résultat attendu n'est pas le même ❌

Cas 8 : choisirColonne = VIDE (soit colonne = VIDE dans la procédure)

```

      X
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
Tapez 'q' pour aller à gauche, 'd' pour aller à droite, espace pour valider
Tapez 'q' pour aller à gauche, 'd' pour aller à droite, espace pour valider

...Program finished with exit code 0
Press ENTER to exit console.

```

Résultat obtenu :

le programme se stop dès que je souhaite choisir la colonne du pion

Résultat du test :

Le test n'est pas réussi car le résultat attendu n'est pas le même ❌

Procédure choisirColonne :

Cas 1 : Tableau = VIDE

[illegible]

Résultat obtenu :

affichage infini du tableau des lignes et des colonnes

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. ❌

Cas 2 : Tableau = de 6 lignes et 7 colonnes

```

      x
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----

Tapez 'q' pour aller à gauche, 'd' pour aller à droite, espace pour valider

```

Résultat obtenu :

Il regarde si tout les case sont pleine ou non (ici non et alors renvoie faux la grille n'est pas pleine)

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Cas 3 : PION_A = 1

```

      o
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----
| | | | | | | |
-----

Tapez 'q' pour aller à gauche, 'd' pour aller à droite, espace pour valider

```

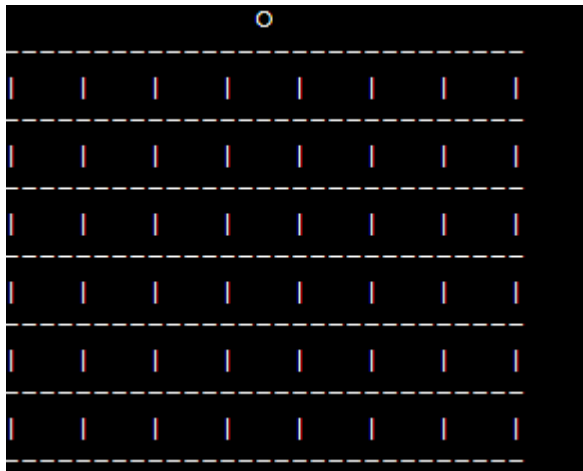
Résultat obtenu :

Il affiche bien le pion A avec un caractère, mais comme 1 n'est pas un caractère alors il affiche rien (on peut le voir sur une case du bas qui est plus petite que les autres)

Résultat du test :

Le test n'est pas réussi car le résultat attendu n'est pas le même ✗

Cas 4 : colonne = 8



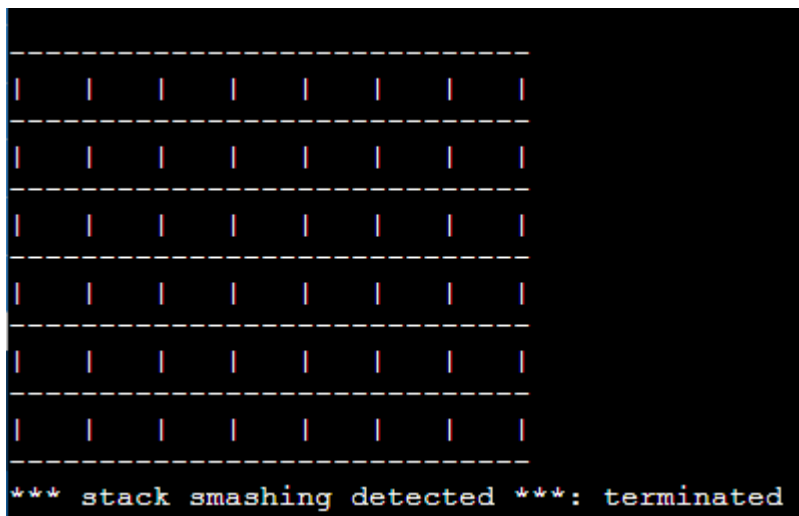
Résultat obtenu :

Il affiche bien le pion dans un case non comprise dans le tableau

Résultat du test :

Le test n'est pas réussi car le résultat attendu n'est pas le même ✗

Cas 5 : colonne_debut = 8



Résultat obtenu :

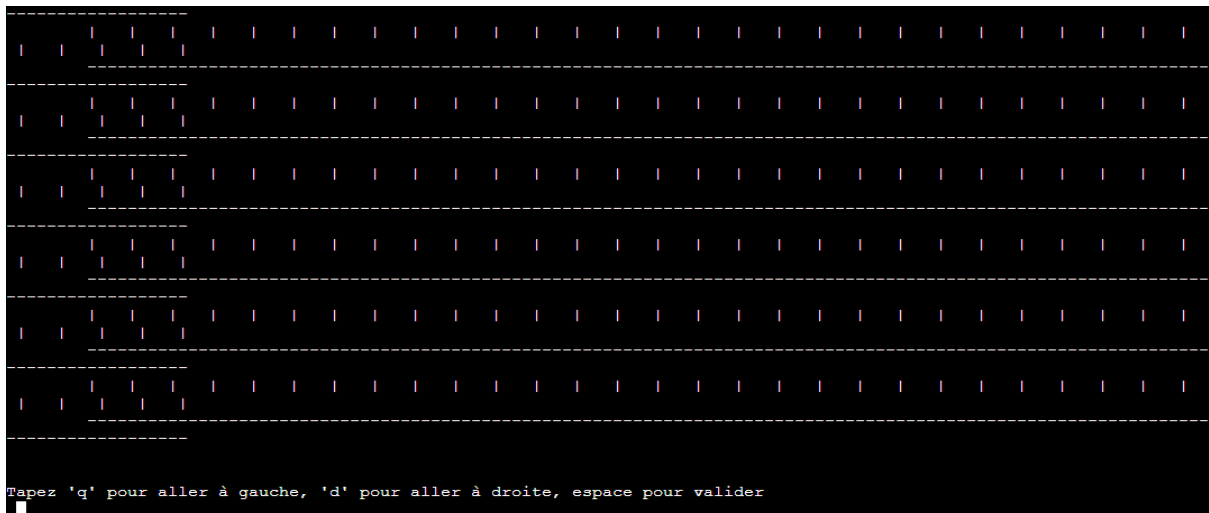
stack smashing detected = erreur dépassement d'un tableau donc la valeur COLONNE_DEBUT = 8 est trop grande pour le tableau au dessus de la grille

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Procédure trouverLigne :

Cas 1 : Tableau = VIDE



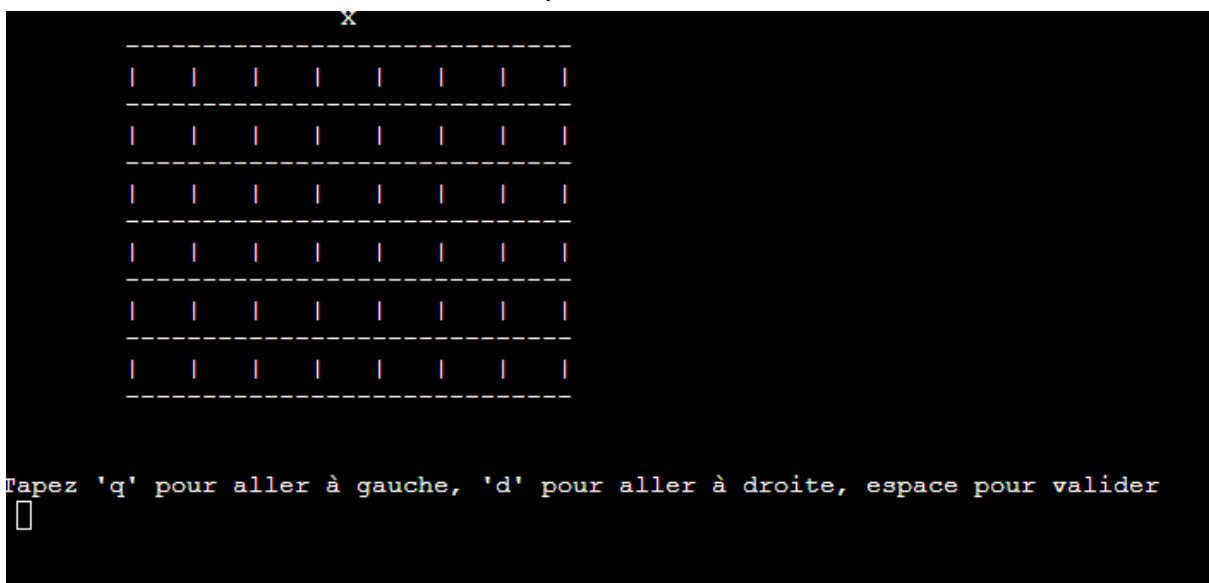
Résultat obtenu :

affichage infini du tableau des lignes et des colonnes

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. ✗

Cas 2 : Grille valide + colonne non pleine



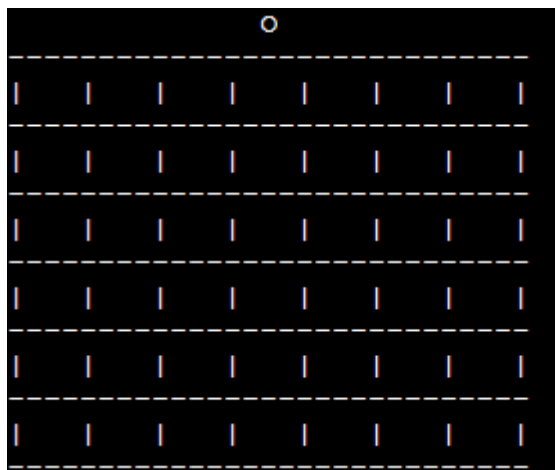
Résultat obtenu :

Il regarde si tout les case sont pleine ou non (ici non et alors renvoie faux la grille n'est pas pleine)

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Cas 3 : Grille valide + ligne non valide (ici ligne =9)



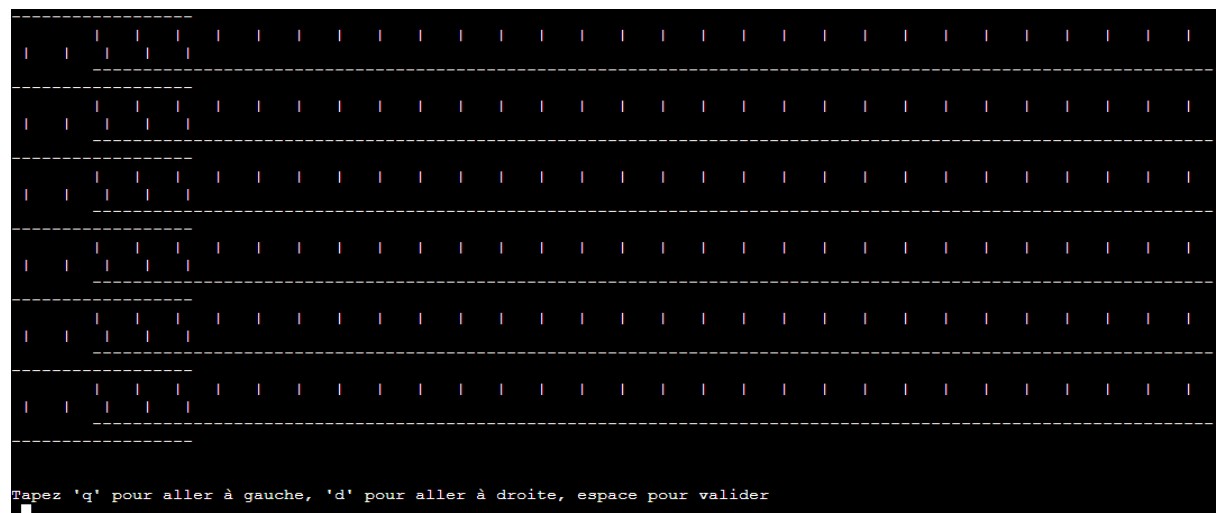
Résultat obtenu :

Le pion c'est placé dans une ligne qui n'est pas visible dans le tableau

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. ❌

Cas 4 : Tableau est vide et la colonne n'est pas pleine



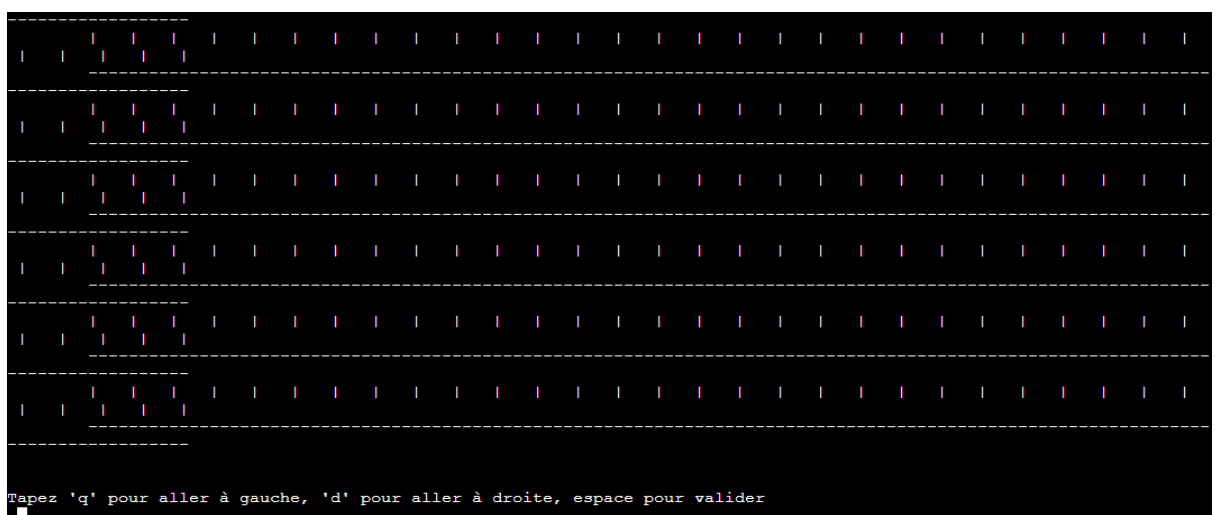
Résultat obtenu :

La grille est affichée mais les pion son placé nulpart

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. ❌

Cas 5 : Tableau est vide et la colonne est pleine



Résultat obtenu :

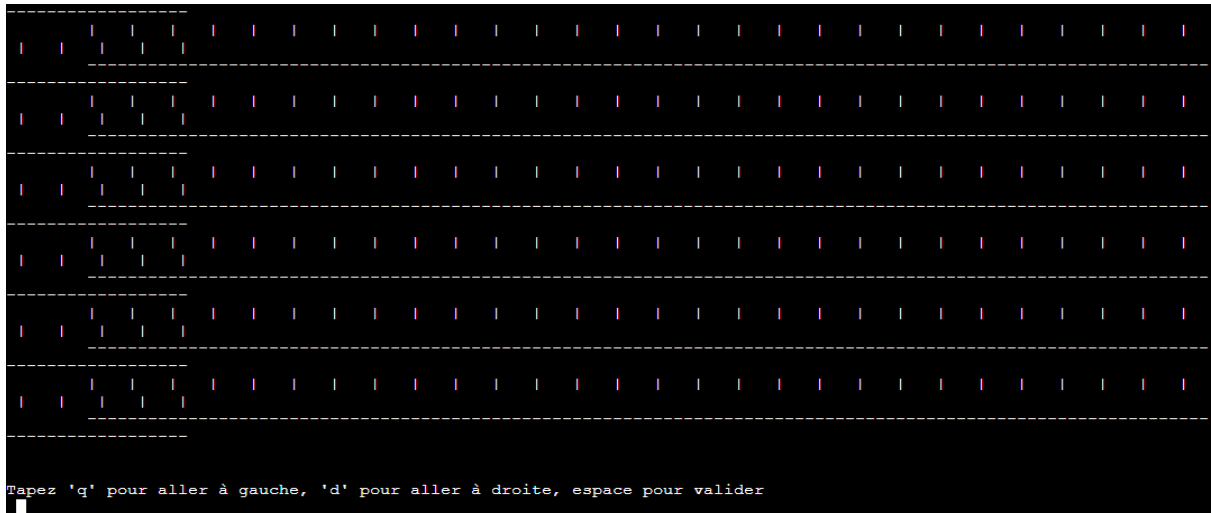
La grille est affichée mais les pion sont placé nulpart

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. ❌

Procédure estVainqueur :

Cas 1 : Tableau = VIDE



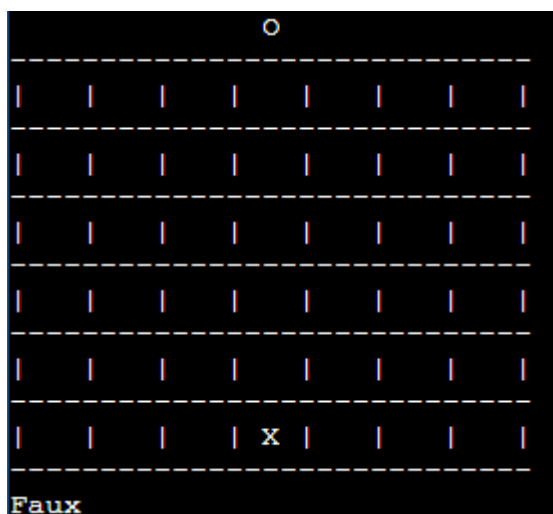
Résultat obtenu :

affichage infini du tableau des lignes et des colonnes

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. ❌

Cas 2 : Grille valide + colonne valide



Résultat obtenu :

Renvoie faux car le pion ne fait pas un puissance 4

Résultat du test : Le test est réussi car le résultat attendu est le même ✅

Cas 3 : Grille valide + ligne valide



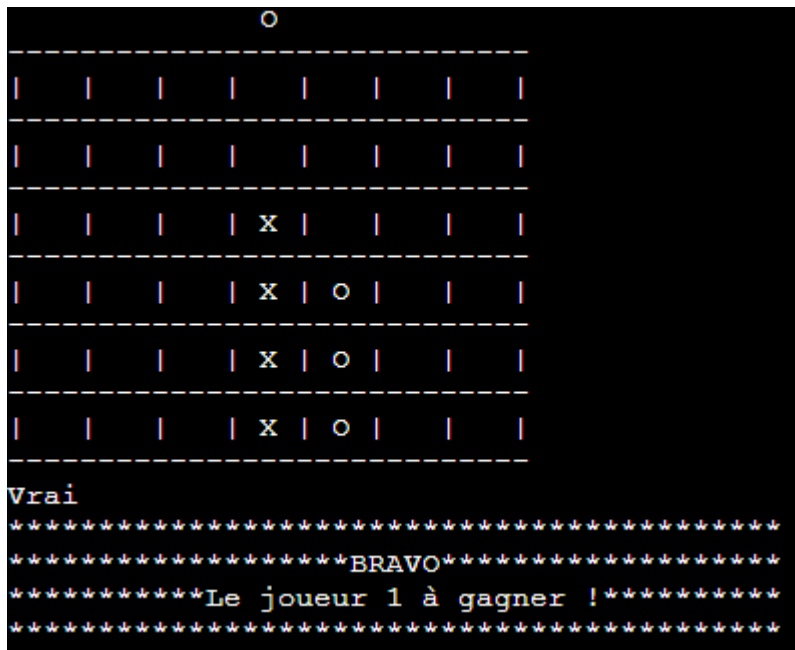
Résultat obtenu :

Le programme affiche bien faux car le pion placé ne fait pas un puissance 4

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Cas 4 : Grille valide + colonne valide



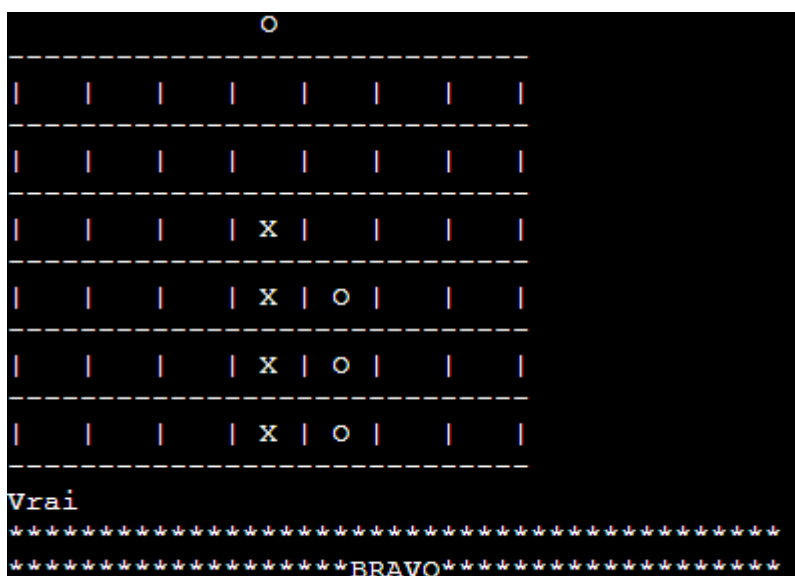
Résultat obtenu :

Le programme affiche bien vrai car le pion placé fait un puissance 4

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Cas 5 : Grille valide + ligne valide



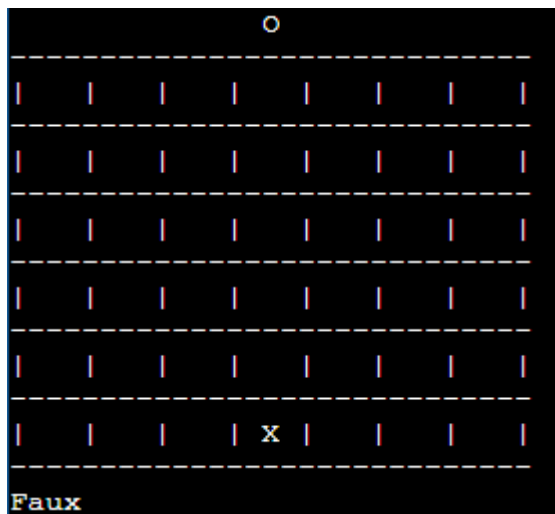
Résultat obtenu :

Le programme affiche bien vrai car le pion placé fait un puissance 4

Résultat du test :

Le test est réussi car le résultat attendu est le même ✓

Cas 6 : Grille valide + colonne valide

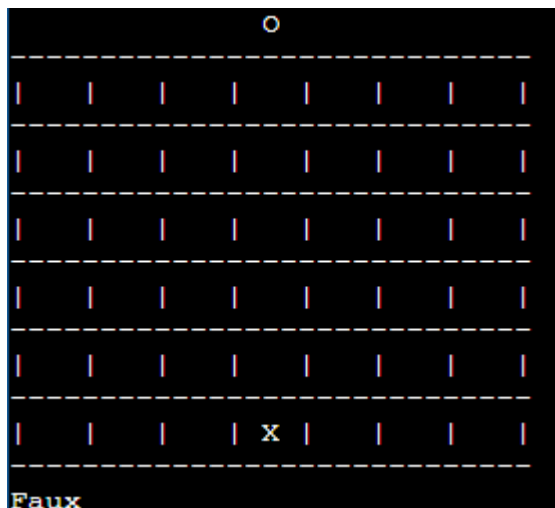


Résultat obtenu :

Renvoie faux car le pion ne fait pas un puissance 4 en diagonale

Résultat du test : Le test est réussi car le résultat attendu est le même ✓

Cas 7 : Grille valide + ligne valide

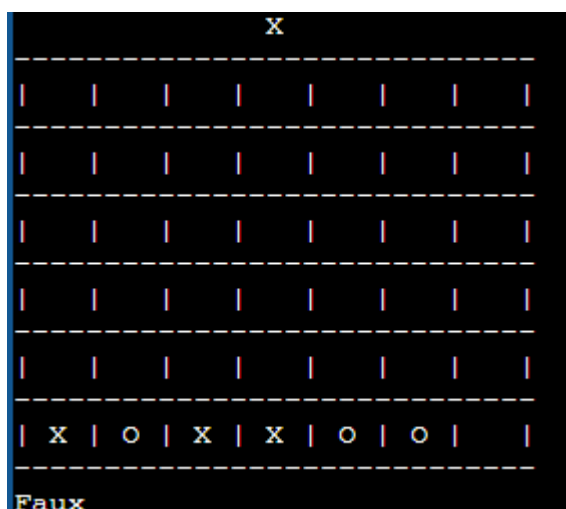


Résultat obtenu :

Renvoie faux car le pion ne fait pas un puissance 4 en diagonale

Résultat du test : Le test est réussi car le résultat attendu est le même ✓

Cas 8 : Grille valide + colonne valide

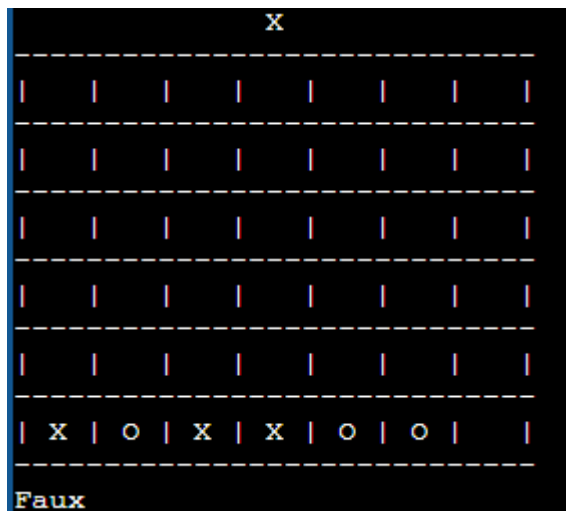


Résultat obtenu :

Renvoie faux car il n'y a que 2 pions alignés

Résultat du test : Le test est réussi car le résultat attendu est le même ✓

Cas 9 : Grille valide + ligne valide

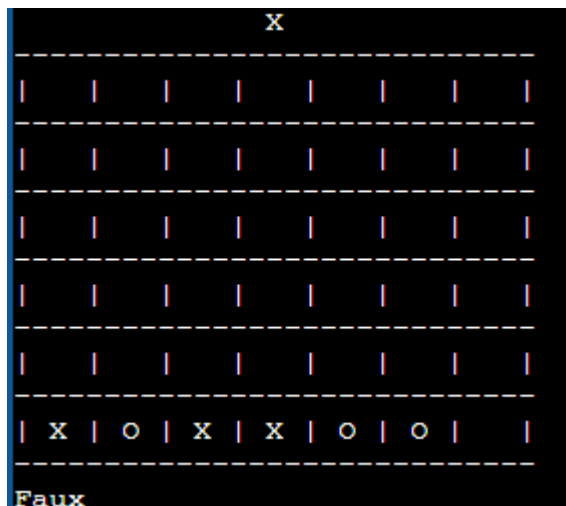


Résultat obtenu :

Renvoie faux car il n'y a que 2 pions alignés

Résultat du test : Le test est réussi car le résultat attendu est le même ✓

Cas 10 : Grille valide + colonne valide

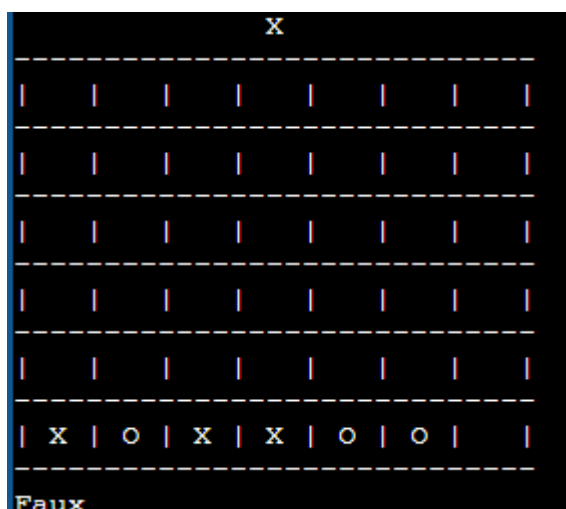


Résultat obtenu :

Renvoie faux car il n'y a que 3 pions alignés

Résultat du test : Le test est réussi car le résultat attendu est le même ✓

Cas 11 : Grille valide + ligne valide



Résultat obtenu :

Renvoie faux car il n'y a que 3 pions alignés

Résultat du test : Le test est réussi car le résultat attendu est le même ✓

Procédure fin de Partie :

Cas 1 : Le pion A gagne

```

      O
-----
|  |  |  |  |  |  |  |  |
-----
|  |  |  |  |  |  |  |
-----
|  |  |  | X |  |  |  |
-----
|  |  |  | X | O |  |  |
-----
|  |  |  | X | O |  |  |
-----
|  |  |  | X | O |  |  |
-----
*****
*****BRAVO*****
*****Le joueur 1 à gagner !*****
*****

```

Résultat obtenu :

Affiche le gagnant et met fin au programme

Résultat du test : Le test est réussi car le résultat attendu est le même ✓

Cas 2 : Le pion A gagne

```

      O
-----
|  |  |  |  |  |  |  |
-----
|  |  |  |  |  |  |  |
-----
|  |  |  | O |  |  |  |
-----
|  |  |  | O | X |  |  |
-----
|  |  |  | O | X |  |  |
-----
|  |  | X | O | X |  |  |
-----
*****
*****BRAVO*****
*****Le joueur 2 à gagner !*****
*****

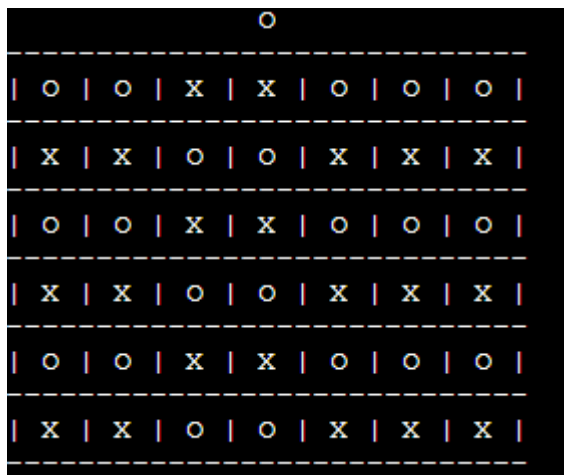
```

Résultat obtenu :

Affiche le gagnant et met fin au programme

Résultat du test : Le test est réussi car le résultat attendu est le même ✓

Cas 3 : Le pion A gagne



Résultat obtenu :

La grille est pleine mais il n'affiche pas qu'il y a égalité et ne met pas fin au programme

Résultat du test :

Le test n'est pas réussi car mon résultat attendu n'est pas le même. ❌