

Cahier de TEST Puissance 4

YANIS PONTYOU

Cahier de TEST Puissance 4	1
FONCTION INIT GRILLE	2
<u>Cas testé : Grille non initialisée ⇒ grille initialisée</u>	2
<u>Cas testé : Grille déjà initialisée ⇒ grille initialisée</u>	2
<u>Cas testé : Grille déjà remplie avec des jetons ⇒ grille initialisée</u>	3
FONCTION AFFICHER	3
<u>Cas testé : Grille vide ⇒ grille affiché sans pion</u>	3
<u>Cas testé : Grille pleine ⇒ grille affiché avec 42 pions</u>	3
<u>Cas testé : Grille partiellement remplie ⇒ grille affiché</u>	4
FONCTION GRILLE PLEINE	4
<u>Cas testé : Grille non pleine ⇒ False</u>	4
<u>Cas testé : Grille pleine ⇒ False</u>	4
FONCTION JOUER	5
<u>Cas testé : on joue dans une colonne non pleine ⇒ ligne et colonne conformes</u>	5
<u>Cas testé : on joue dans une colonne pleine ⇒ invite à choisir une autre colonne</u>	6
FONCTION CHOISIR COLONNE	8
<u>Cas testé : pour le déplacement : caractère saisi non valide ⇒ redemande la saisie</u>	8
<u>Cas testé : colonne entre 2 et 7 et saisie de 'q' ⇒ pion déplacé vers la gauche (n° de colonne décrémenté)</u>	8
<u>Cas testé : colonne entre 1 et 6 et saisie de 'd' ⇒ pion déplacé vers la droite (n° colonne incrémenté)</u>	9
<u>Cas testé : colonne = 1 et saisie de 'q' ⇒ rien (ou bien boucle en colonne 7)</u>	9
<u>Cas testé : colonne = 7 et saisie de 'd' ⇒ rien (ou bien boucle en colonne 1)</u>	10
<u>Cas testé : colonne valide et saisie de la commande pour faire tomber le pion (touche ESPACE dans les specs) ⇒ retourne le n° de colonne courant</u>	10
FONCTION TROUVER LIGNE	13
<u>Cas testé : colonne pleine ⇒ -1</u>	13
<u>Cas testé : colonne non pleine ⇒ le n° ligne correspond à la première case libre de la colonne</u>	14
FONCTION estVainqueur	14
<u>Cas testé : pas 4 pions identiques alignés ⇒ False</u>	14
<u>Cas testé : colonne d'au moins 4 pions ⇒ True</u>	15
<u>Cas testé : ligne d'au moins 4 pions ⇒ True</u>	16
<u>Cas testé : diagonale descendante d'au moins 4 pions ⇒ True</u>	17
<u>Cas testé : diagonale ascendante d'au moins 4 pions ⇒ True</u>	19
<u>Cas testé : ligne d'au moins 4 pions interrompue par un pion adverse ⇒ FALSE</u>	20
FONCTION finDePartie	21
<u>Cas testé : pion en entrée = PION A ⇒ affiche "X vainqueur"</u>	21
<u>Cas testé : pion en entrée = PION B ⇒ affiche "O vainqueur"</u>	21
<u>Cas testé : pion en entrée = PION VIDE ⇒ affiche "Match NULL"</u>	22

FONCTION INIT GRILLE

Cas testé : Grille non initialisée \Rightarrow grille initialisée

```

1  int main()
2  {
3      grille tab;
4
5      pinitgrille(tab);
6
7      afficher(tab, PION_A, COLONNE_DEBUT);
8  }
9

```

[illegible]

Cas testé : Grille déjà initialisée \Rightarrow grille initialisée

```

1  int main()
2  {
3      grille tab;
4
5      pinitgrille(tab);
6
7      pinitgrille(tab);
8
9      afficher(tab, PION_A, COLONNE_DEBUT);
10 }
```

1	2	3	4	5	6	7
			X			
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

Cas testé : Grille déjà remplie avec des jetons \Rightarrow grille initialisée

```
1  int main()
2  {
3      grille tab = {'X', 'O'};
4      pinitgrille(tab);
5      afficher(tab, PION_A, COLONNE_DEBUT);
6  }
7
```

	1	2	3	4	5	6	7
X							
	-	-	-	-	-	-	-
	-	-	-	-	-	-	-
	-	-	-	-	-	-	-
	-	-	-	-	-	-	-
	-	-	-	-	-	-	-

FONCTION AFFICHER

Cas testé : Grille vide \Rightarrow grille affiché sans pion

```
1 int main()
2 {
3     grille tab;
4
5     pinitgrille(tab);
6
7     afficher(tab, PION_A, COLONNE_DEBUT);
8 }
```

1	2	3	4	5	6	7
			X			
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

Cas testé : Grille pleine \Rightarrow grille affiché avec 42 pions

```
1 int main()
2 {
3     grille tab;
4     pinitgrille(tab);
5
6     for(int i=0; i<NBLIG; i++){
7         for(int j=0; j<NBCOL; j++){
8             tab[i][j] = PION_B;
9         }
10    }
11
12    afficher(tab, PION_A, COLONNE_DEBUT);
13 }
```

1	2	3	4	5	6	7
			X			
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Cas testé : Grille partiellement remplie \Rightarrow grille affiché

```

1  int main()
2  {
3      grille tab;
4      pinitgrille(tab);
5
6      for(int i=0; i<3; i++){
7          for(int j=0; j<3; j++){
8              tab[i][j] = PION_B;
9          }
10     }
11
12     afficher(tab, PION_A, COLONNE_DEBUT);
13 }

```

1	2	3	4	5	6	7
			X			
0	0	0	-	-	-	-
0	0	0	-	-	-	-
0	0	0	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

FONCTION GRILLE PLEINE

Cas testé : Grille non pleine \Rightarrow False

```

1  int main()
2  {
3      grille tab;
4      pinitgrille(tab);
5
6      tab[5][1] = PION_B;
7      tab[5][2] = PION_A;
8
9      if (grillePleine(tab)==false){
10         printf("False");
11     }
12     else{
13         printf("True");
14     }
15
16     afficher(tab, PION_A, COLONNE_DEBUT);
17 }

```

False

'clear' n'est pas reconnu en tant que commande interne, externe, un programme exécutable ou un fichier de commandes.

1	2	3	4	5	6	7
			X			
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	0	X	-	-	-	-

Cas testé : Grille pleine \Rightarrow False

```

1  int main()
2  {
3      grille tab;
4      pinitgrille(tab);
5
6      for(int i=0; i<NBLIG; i++){
7          for(int j=0; j<NBCOL; j++){
8              tab[i][j] = PION_B;
9          }
10     }
11
12     if (grillePleine(tab)==false){
13         printf("False \n");
14     }
15     else{
16         printf("True \n");
17     }
18
19     afficher(tab, PION_A, COLONNE_DEBUT);
20 }

```

```

True
'clear' n'est pas reconnu
ou externe, un programme
1 2 3 4 5 6 7
  X

0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0

```

FONCTION JOUER

Cas testé : on joue dans une colonne non pleine \Rightarrow ligne et colonne conformes

```

1 int main()
2 {
3     // Déclaration des variables
4     char prenom1[20], prenom2[20];
5     int ligne, colonne;
6     char winner, tmp;
7     grille tab;
8
9     // Etape de la saisie des prénoms:
10    printf("Le jeu du puissance 4 \n");
11    //Cf maquette :)
12    printf("Prénom du joueur 1 : \n");
13    scanf("%s", prenom1, &tmp);
14    printf("Prénom du joueur 2 : \n");
15    scanf("%s", prenom2, &tmp);
16
17    //Première étapes d'initialisation
18    pinitgrille(tab);
19    winner = VIDE;
20    afficher(tab, PION_A, COLONNE_DEBUT);
21
22    while (winner!=INCONNU && (grillePleine(tab)==false))
23    {
24        jouer(tab, PION_A, &ligne, &colonne);
25        afficher(tab, PION_B, COLONNE_DEBUT);
26        if (estVainqueur(tab, ligne, colonne))
27        {
28            winner = PION_A;
29        }
30        else if (grillePleine(tab)==false)
31        {
32            jouer(tab, PION_B, &ligne, &colonne);
33            afficher(tab, PION_A, COLONNE_DEBUT);
34            if (estVainqueur(tab, ligne, colonne))
35            {
36                winner = PION_B;
37            }
38        }
39        finDePartie(winner, prenom1, prenom2);
40    }
41

```

```

1 2 3 4 5 6 7
0
- - - - -
- - - - -
- - - - -
- - - - -
- - X - -
- 0 X - -

```

Pour vous déplacer veuillez appuyer sur :

q + [entrée]: Pour aller à gauche

d + [entrée]: pour aller à droite

[espace] + [entrée]: pour valider

'clear' n'est pas reconnu en tant que commande interne, un programme exécutable ou un fichier de commandes.

```

1 2 3 4 5 6 7
X
- - - - -
- - - - -
- - - - -
- - 0 - -
- - X - -
- 0 X - -

```

Pour vous déplacer veuillez appuyer sur :

q + [entrée]: Pour aller à gauche

d + [entrée]: pour aller à droite

[espace] + [entrée]: pour valider

□

```

1 void jouer(grille g, char pion, int *ligne, int *colonne)
2 {
3
4     (*colonne) = choisirColonne(g, pion, COLONNE_DEBUT); // Permet de choisir la colonne
5
6     (*ligne) = trouverLigne(g, *colonne); // Permet de trouver dans la colonne qu'elle ligne est disponible
7
8     g[*ligne][*colonne] = pion; // Place le pion au bon endroit
9
10 }

```

Cas testé : on joue dans une colonne pleine⇒ invite à choisir une autre colonne

```

1  int choisirColonne(grille g, char pion,
   int colonne)
2  {
3      int col = colonne;
4      char deplacement, tmp;
5
6
7      printf("Pour vous déplacer veuillez
   appuyer sur : \n q + [entrée]: Pour aller
   à gauche \n d + [entrée]: pour aller
   à droite \n[espace] + [entrée]: pour valider \n"); // Message d'instruction
8      scanf("%c%c", &deplacement, &tmp);
   // Lire la touche en faisant attention
   à la touche [entrée]
9
10
11     while ((trouverLigne(g, col) == -1)
   || (deplacement != VIDE)) // Si VIDE =
   la touche [espace] donc il faut arrêter
   la boucle Sinon si -1 c'est que la colonne
   est pleine
12     {
13         switch (deplacement) //Un switch
   pour optimiser la chose sachant que
   l'on a 3 cas " " : placer le jeton "q"
   : gauche"d" : droite
14         {
15             case ' ':
16                 if(trouverLigne(g, col) ==
   -1) // Si -1 la colonne est pleine
17                 {
18                     printf("La colonne est
   malheureusement pleine");
19                 }
20                 break;

```

```

1  void jouer(grille g, char pion, int *Ligne,
   int *colonne)
2  {
3
4      (*colonne) = choisirColonne(g, pion,
   COLONNE_DEBUT); // Permet de choisir
   la colonne
5
6      (*Ligne) = trouverLigne(g, *colonne);
   // Permet de trouver dans la colonne
   qu'elle ligne est disponible
7
8      g[*Ligne][*colonne] = pion; // Placer
   le pion au bon endroit
9
10 }

```

```

1  2  3  4  5  6  7
      X
- - - - -
- - - 0 - - -
- - - X - - -
- - - 0 - - -
- - - X - - -
- - - 0 - - -
- - - X - - -
- - - 0 X - - -

Pour vous déplacer veuillez appuyer sur :
q + [entrée]: Pour aller à gauche
d + [entrée]: pour aller à droite
[espace] + [entrée]: pour valider

La colonne est malheureusement pleine 'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.
1  2  3  4  5  6  7
      0
- - - X - - -
- - - 0 - - -
- - - X - - -
- - - 0 - - -
- - - X - - -
- - - 0 X - - -

```

FONCTION CHOISIR COLONNE

Cas testé : pour le déplacement : caractère saisi non valide \Rightarrow redemande la saisie

```
 1 2 3 4 5 6 7
      X

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

Pour vous déplacer veuillez appuyer sur :
q + [entrée]: Pour aller à gauche
d + [entrée]: pour aller à droite
[espace] + [entrée]: pour valider
R
Vous avez surement tapez un mauvais caractère pour rapelle :
[espace] + [entrée]: pour valider
'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.
 1 2 3 4 5 6 7
      X

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

█
```

Cas testé : colonne entre 2 et 7 et saisie de 'q' \Rightarrow pion déplacé vers la gauche (n° de colonne décrémenté)

\Rightarrow

\Rightarrow



Cas testé : colonne entre 1 et 6 et saisie de 'd' ⇒ pion déplacé vers la droite
(n° colonne incrémenté)

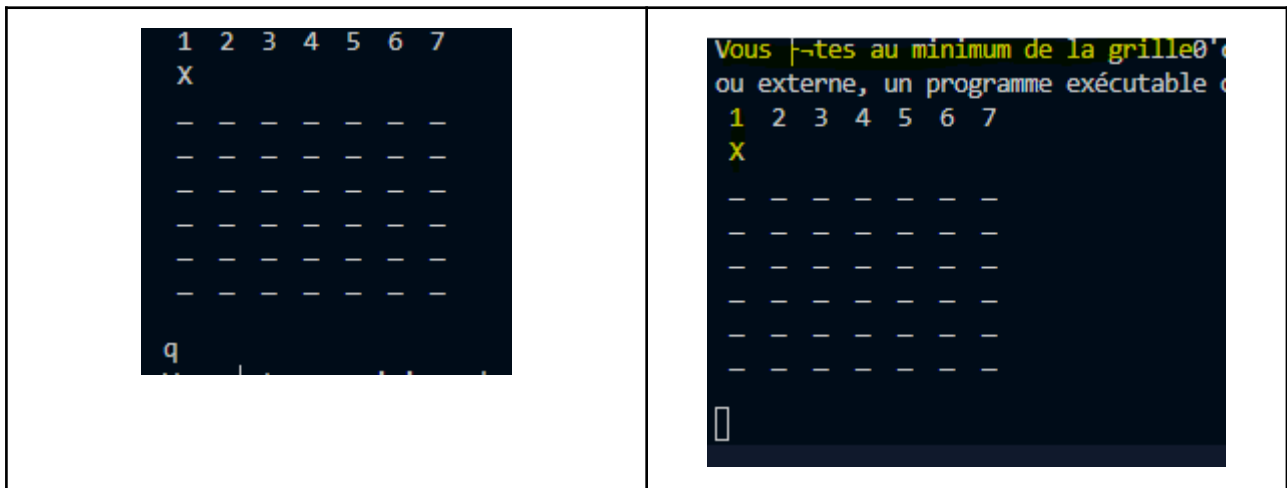
⇒

⇒



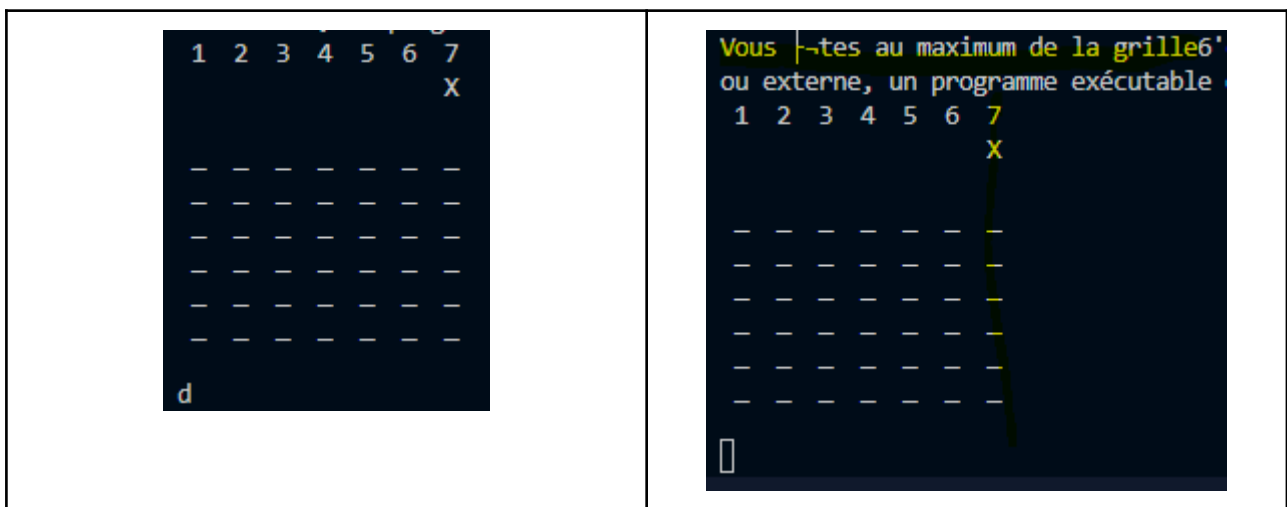
Cas testé : colonne = 1 et saisie de 'q' ⇒ rien (ou bien boucle en colonne 7)

⇒



Cas testé : colonne = 7 et saisie de 'd' ⇒ rien (ou bien boucle en colonne 1)

⇒



Cas testé : colonne valide et saisie de la commande pour faire tomber le pion (touche ESPACE dans les specs) ⇒ retourne le n° de colonne courant

```

1 2 3 4 5 6 7
      X

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

Pour vous déplacer veuillez appuyer sur :
q + [entrée]: Pour aller à gauche
d + [entrée]: pour aller à droite
[espace] + [entrée]: pour valider

```

```

[espace] <3>
'clear' n'est pas reconnu en tant que commande
ou externe, un programme exécutable ou un fichier
1 2 3 4 5 6 7
      0

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

- - - X - - -

Pour vous déplacer veuillez appuyer sur :
q + [entrée]: Pour aller à gauche
d + [entrée]: pour aller à droite
[espace] + [entrée]: pour valider

```



```
1  int choisirColonne(grille g, char pion, int colonne)
2  {
3      int col = colonne;
4      char deplacement, tmp;
5
6
7      printf("Pour vous déplacer veuillez appuyer sur : \n q + [entrée]: Pour aller à gauche \n d + [entrée]: pour aller à droite \n[espace] + [entrée]: pour valider \n"); // Message d'instruction
8      scanf("%c%c", &deplacement, &tmp); // Lire la touche en faisant attention à la touche [entrée]
9
10
11     while ((trouverLigne(g, col) == -1) || (deplacement != VIDE)) // Si VIDE = La touche [espace] donc il faut arrêter la boucle Sinon si -1 c'est que la colonne est pleine
12     {
13         switch (deplacement) //Un switch pour optimiser la chose sachant que l'on a 3 cas " " : placer le jeton "q" : gauche "d" : droite
14         {
15             case ' ':
16                 if(trouverLigne(g, col) == -1) // Si -1 la colonne est pleine
17                 {
18                     printf("La colonne est malheureusement pleine");
19                 }
20                 break;
21
22             case 'd':
23                 if(col < NBCOL -1)
24                 {
25                     col = col +1;
26                 }
27                 else{
28                     printf("Vous êtes au maximum de la grille");
29                 }
30                 break;
31
32             case 'q':
33                 if(col > 0)
34                 {
35                     col = col -1;
36                 }
37                 else
38                 {
39                     printf("Vous êtes au minimum de la grille");
40                 }
41                 break;
42
43             default:
44                 printf("Vous avez sûrement tapé un mauvais caractère pour rappelez : \n q + [entrée]: Pour aller à gauche \n d + [entrée]: pour aller à droite \n[espace] + [entrée]: pour valider \n");
45                 break;
46         }
47
48
49         afficher(g, pion, col);
50         scanf("%c%c", &deplacement, &tmp);
51     }
52     printf("\n [espace ] <%d> \n", col);
53     return col;
54 }
```

FONCTION TROUVER LIGNE

Cas testé : colonne pleine \Rightarrow -1

```
1 int trouverLigne(grille g, int colonne)
2 {
3     int l;
4     l = -1; //Initialisation à -1 car s
5     i le cas ou la colonne est pleine sinon
6     on aurait mis 0
7
8     while (g[l+1][colonne] == VIDE && l
9     < NBLIG) //Test si la case est vide ou
10    sinon si on a parcouru toute la colonn
11 e {
12     l++;
13 }
14 print(" \n %d \n", l);
15 return l;
16 }
```

```
0
- - - - -
- - - X - - -
- - - 0 - - -
- - - X - - -
- - - 0 - - -
- - - X - - -

Pour vous déplacer veuillez
q + [entrée]: Pour aller
d + [entrée]: pour aller
[espace] + [entrée]: pour v

0
0
-1
La colonne est malheureusement
ou externe, un programme exé
1 2 3 4 5 6 7
      X

- - - 0 - - -
- - - X - - -
- - - 0 - - -
- - - X - - -
- - - 0 - - -
- - - X - - -
```

Cas testé : colonne non pleine \Rightarrow le n° ligne correspond à la première case libre de la colonne

```

1  int trouverLigne(grille g, int colonne)
2  {
3      int l;
4      l = -1; //Initialisation à -1 car s
5      i le cas où la colonne est pleine sinon
6      on aurait mis 0
7
8      while (g[l+1][colonne] == VIDE && l
9      < NBLIG) //Test si la case est vide ou
10     sinon si on a parcouru toute la colonn
11     e {
12         l++;
13     }
14     print("\n %d \n", l);
15     return l;
16 }

```

```
PS C:\Users\moi\Desktop\>
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

Pour vous déplacer veuillez utiliser :
q + [entrée] : Pour aller en arrière
d + [entrée] : pour aller en avant
[espace] + [entrée] : pour aller à la ligne suivante

5
5
'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme externe n'a pu être exécuté.
1 2 3 4 5 6 7
0
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
```

FONCTION estVainqueur

Cas testé : pas 4 pions identiques alignés \Rightarrow False



1	2	3	4	5	6	7
---	---	---	---	---	---	---

'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier

```
1 2 3 4 5 6 7
0
```

```
-- -- -- -- --
-- -- -- -- --
-- -- -- -- --
-- -- 0 X -- --
-- -- 0 X -- --
-- -- 0 X -- --
-- -- 0 X -- --
```

FALSE Pour vous déplacer veuillez appuyer sur :

q + [entrée]: Pour aller à gauche

d + [entrée]: pour aller à droite

[espace] + [entrée]: pour valider

```
1 2 3 4 5 6 7
X
```

```
-- -- -- -- --
-- -- -- -- --
-- -- 0 -- -- --
-- -- 0 X -- --
-- -- 0 X -- --
-- -- 0 X -- --
-- -- 0 X X -- --
```

'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier

```
1 2 3 4 5 6 7
0
```

```
-- -- -- -- --
-- -- -- -- --
-- -- 0 X -- --
-- -- 0 X -- --
-- -- 0 X -- --
-- -- 0 X -- --
-- -- 0 X X -- --
```

TRUE Pour vous déplacer veuillez appuyer sur :

q + [entrée]: Pour aller à gauche

d + [entrée]: pour aller à droite

[espace] + [entrée]: pour valider

□

Cas testé : ligne d'au moins 4 pions ⇒ True


```

q
'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.

1 2 3 4 5 6 7
X

- - - - -
- - - - -
- - - 0 - - -
- - - 0 - - -
- - - 0 - - -
- X X X - - -

q
'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.

1 2 3 4 5 6 7
X

- - - - -
- - - - -
- - - 0 - - -
- - - 0 - - -
- - - 0 - - -
- X X X - - -

'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.

1 2 3 4 5 6 7
0

- - - - -
- - - - -
- - - 0 - - -
- - - 0 - - -
- - - 0 - - -
X X X X - - -

TRUE
Pour vous déplacer veuillez appuyer sur le clavier
q + [entrée]: Pour aller à gauche
d + [entrée]: pour aller à droite
[espace] + [entrée]: pour valider

```

Cas testé : diagonale descendante d'au moins 4 pions ⇒ True

Pour vous déplacer veuillez appuyer sur :

q + [entr|e]: Pour aller à gauche

d + [entr|e]: pour aller à droite

[espace] + [entr|e]: pour valider

d

'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier

```
1 2 3 4 5 6 7
          X
```

```
- - - - -
- - - - -
- - - - -
- - - 0 X - -
- - - 0 X X -
- - 0 X 0 0 X
```

q

'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier

```
1 2 3 4 5 6 7
          X
```

```
- - - - -
- - - - -
- - - - -
- - - 0 X - -
- - - 0 X X -
- - 0 X 0 0 X
```

'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier

```
1 2 3 4 5 6 7
          0
```

```
- - - - -
- - - - -
- - - X - - -
- - - 0 X - -
- - - 0 X X -
- - 0 X 0 0 X
```

TRUEPour vous déplacer veuillez appuyer sur :

q + [entr|e]: Pour aller à gauche

d + [entr|e]: pour aller à droite

[espace] + [entr|e]: pour valider

Cas testé : diagonale ascendante d'au moins 4 pions ⇒ True

```
PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL

-- -- -- -- -- X 0
-- -- -- X X 0 0
-- -- -- X 0 X 0

d
'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commande
1 2 3 4 5 6 7
      X

-- -- -- -- -- --
-- -- -- -- -- --
-- -- -- -- -- --
-- -- -- -- X 0
-- -- -- X X 0 0
-- -- -- X 0 X 0

d
'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commande
1 2 3 4 5 6 7
      X

-- -- -- -- -- --
-- -- -- -- -- --
-- -- -- -- -- --
-- -- -- -- X 0
-- -- -- X X 0 0
-- -- -- X 0 X 0

'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commande
1 2 3 4 5 6 7
      0

-- -- -- -- -- --
-- -- -- -- -- --
-- -- -- -- -- X
-- -- -- -- X 0
-- -- -- X X 0 0
-- -- -- X 0 X 0

TRUEPour vous déplacer veuillez appuyer sur :
q + [entrée]: Pour aller à gauche
d + [entrée]: pour aller à droite
[espace] + [entrée]: pour valider
█
```

Cas testé : ligne d'au moins 4 pions interrompue par un pion adverse ⇒
FALSE

```
puissance4.c > main()
460      jouer(tab, PION_A, &ligne, &colonne);
461      afficher(tab, PION_B, COLONNE_DEBUT, COLONNE_FIN);
462      if (estVainqueur(tab, ligne, colonne))
      return 1;

PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL

  _ _ 0 X 0 X _
d
'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.
1 2 3 4 5 6 7
      X

  _ _ _ _ _ _ _
  _ _ _ _ _ _ _
  _ _ _ _ _ _ _
  _ _ _ _ _ _ _
  _ _ 0 X 0 X _
  _ _ _ _ _ _ _

d
'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.
1 2 3 4 5 6 7
      X

  _ _ _ _ _ _ _
  _ _ _ _ _ _ _
  _ _ _ _ _ _ _
  _ _ _ _ _ _ _
  _ _ 0 X 0 X _
  _ _ _ _ _ _ _

'clear' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.
1 2 3 4 5 6 7
      0

  _ _ _ _ _ _ _
  _ _ _ _ _ _ _
  _ _ _ _ _ _ _
  _ _ _ _ _ _ _
  _ _ 0 X 0 X X
  _ _ _ _ _ _ _

FALSE Pour vous déplacer veuillez appuyer sur :
q + [entr]: Pour aller à gauche
d + [entr]: Pour aller à droite
```

FONCTION finDePartie

Cas testé : pion en entrée = PION A ⇒ affiche "X vainqueur"

```
PS C:\Users\moi\Desktop\Cahier_test2> .\a.exe
Le jeu du puissance 4
Prénom du joueur 1 :
Jean_Cristophe_Vialat
Prénom du joueur 2 :
Yanis :)
Victoire du joueur Jean_Cristophe_Vialat avec les pions 'X' Bravo à toi !
```

```
1  int main()
2  {
3      // Déclaration des variables
4      char prenom1[20], prenom2[20];
5      char tmp;
6
7      grille tab;
8
9      // Etape de la saisie des prénoms:
10     printf("Le jeu du puissance 4 \n");
11     //Cf maquette :)
12     printf("Prénom du joueur 1 : \n");
13     scanf("%s%c", prenom1, &tmp);
14     printf("Prénom du joueur 2 : \n");
15     scanf("%s%c", prenom2, &tmp);
16
17     finDePartie(PION_A, prenom1, prenom2);
18 }
```

Cas testé : pion en entrée = PION B ⇒ affiche "O vainqueur"

```

1  int main()
2  {
3      // Déclaration des variables
4      char prenom1[20], prenom2[20];
5      char tmp;
6
7      grille tab;
8
9      // Etape de la saisie des prénoms:
10     printf("Le jeu du puissance 4 \n");
11     //Cf maquette :)
12     printf("Prénom du joueur 1 : \n");
13     scanf("%s%c", prenom1, &tmp);
14     printf("Prénom du joueur 2 : \n");
15     scanf("%s%c", prenom2, &tmp);
16
17     finDePartie(PION_B, prenom1, prenom
2);
17 }

```

```

PS C:\Users\moi\Desktop\Cahier_test2> .\a.exe
Le jeu du puissance 4
Prénom du joueur 1 :
Yanis :)
Prénom du joueur 2 :
Jean_Cristophe_Vialat
Victoire du joueur Jean_Cristophe_Vialat avec les pions 'O' Bravo à toi !
PS C:\Users\moi\Desktop\Cahier_test2> 

```

Cas testé : pion en entrée = PION VIDE ⇒ affiche "Match NULL"

```

1  int main()
2  {
3      // Déclaration des variables
4      char prenom1[20], prenom2[20];
5      char tmp;
6
7      grille tab;
8
9      // Etape de la saisie des prénoms:
10     printf("Le jeu du puissance 4 \n");
11     //Cf maquette :)
12     printf("Prénom du joueur 1 : \n");
13     scanf("%s%c", prenom1, &tmp);
14     printf("Prénom du joueur 2 : \n");
15     scanf("%s%c", prenom2, &tmp);
16
17     finDePartie(VIDE, prenom1, prenom2
18 );
19 }

```

```

PS C:\Users\moi\Desktop\Cahier_test2> .\a.exe
Le jeu du puissance 4
Prénom du joueur 1 :
Yanis
Prénom du joueur 2 :
JEAN_CRISTOPHE_VIALAT
Le vainqueur est ...Egalité !!! Personne ne gagne
PS C:\Users\moi\Desktop\Cahier_test2> 

```