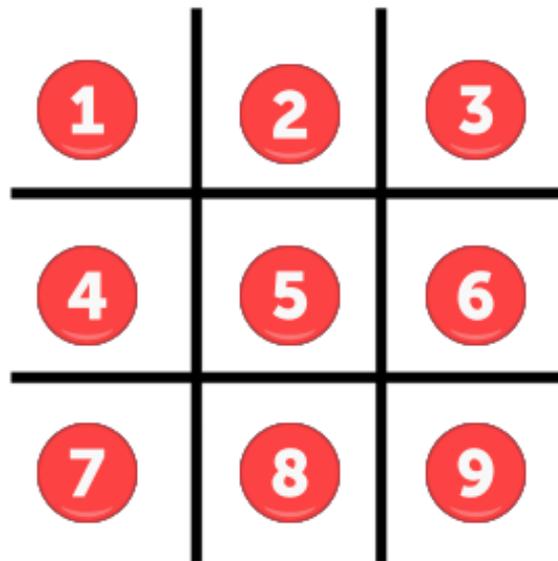


# Tableaux en Java

## Défi n°1: Tic Tac Toe

**Objectif** : Ecrire un programme Java permettant à 2 joueurs de jouer au tic tac toe.

Le programme devra donc être séparé en plusieurs fonctions. A chaque fois que l'on demande à l'utilisateur de rentrer un choix de case, l'utilisateur doit sélectionner une case entre 1 et 9. La position de chaque valeur entre 1 et 9 dans le jeu est comme dans la figure ci-dessous :



Le déroulement du jeu lors de l'exécution du programme devra s'effectuer comme ci-dessous. Le même affichage doit être respecté.

Vous devez faire attention aux contraintes suivantes :

- Vous devez empêcher un joueur de choisir le même emplacement (case) que le joueur précédent.
- Seules les entrées entre 1 et 9 sont autorisées.
- Chaque jeu pourrait avoir le résultat suivant :
  - Le joueur 1 a gagné
  - Le joueur 2 a gagné
  - C'est un match nul, personne n'a gagné
- Pour créer ce mini-jeu, pensez à utiliser des tableaux 1D/2D que vous avez vus dans les TD. N'utilisez pas de structures ou de fonctions prédéfinies que vous n'avez pas vues dans ce cours.

```

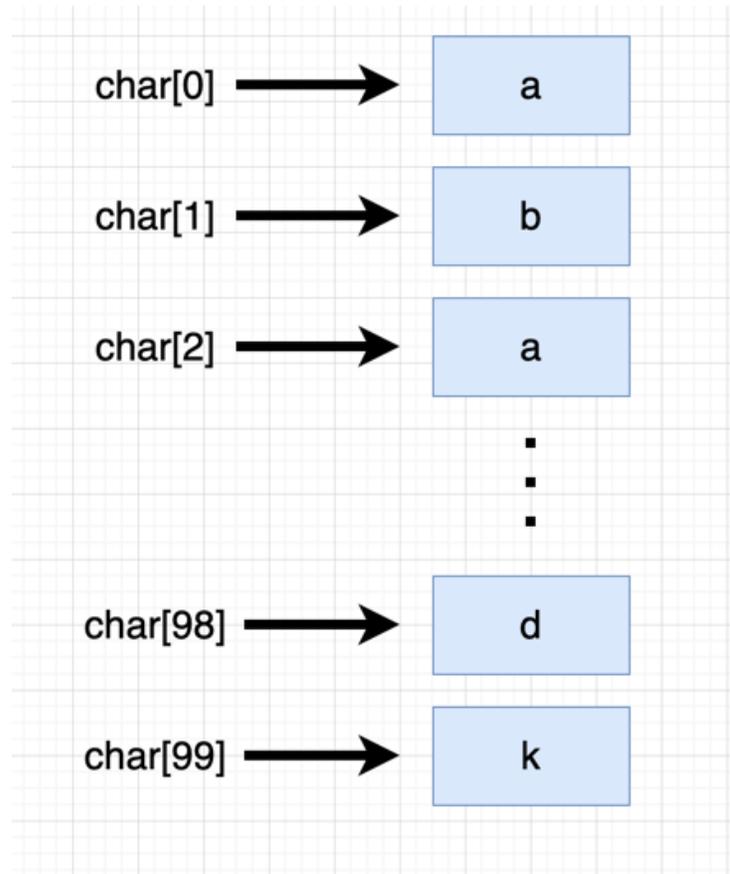
Bienvenue sur Tic Tac Toe
Joueur 1 = X : Joueur 2 = 0
  |  |
- + - + -
  |  |
- + - + -
  |  |
Joueur 1: choisissez votre emplacement (1-9) : 1
X |  |
- + - + -
  |  |
- + - + -
  |  |
=====
Joueur 2: choisissez votre emplacement (1-9) : 3
X |  | 0
- + - + -
  |  |
- + - + -
  |  |
=====
Joueur 1: choisissez votre emplacement (1-9) : 5
X |  | 0
- + - + -
  | X |
- + - + -
  |  |
=====
Joueur 2: choisissez votre emplacement (1-9) : 6
X |  | 0
- + - + -
  | X | 0
- + - + -
  |  |
=====
Joueur 1: choisissez votre emplacement (1-9) : 9
X |  | 0
- + - + -
  | X | 0
- + - + -
  |  | X
=====
Félicitations, le joueur 1 a gagné :)

Process finished with exit code 0

```

## Défi n°2: Compter les occurrences de chaque lettre

Générez **100 lettres minuscules** au hasard et attribuez-les à un tableau de caractères, comme indiqué ci-dessous :



Vous pouvez obtenir une lettre aléatoire en définissant une méthode **randomLettreMinuscule()** par exemple.

### À faire:

1. Comptez les occurrences de chaque lettre dans le tableau. La sortie devrait ressembler à ce qui suit :

Les lettres minuscules sont :

```
e y l s r i b k j v j h a b z n w b t v  
s c c k r d w a m p w v u n q a m p l o  
a z g d e g f i n d x m z o u l o z j v  
h w i w n t g x w c d o t x h y v z y z  
q e a m f w p g u q t r e n n w f c r f
```

Les occurrences de chaque lettre sont :

```
5 a 3 b 4 c 4 d 4 e 4 f 4 g 3 h 3 i 3 j  
2 k 3 l 4 m 6 n 4 o 3 p 3 q 4 r 2 s 4 t  
3 u 5 v 8 w 3 x 3 y 6 z
```

- Affiche la lettre qui apparaît le moins et la lettre qui apparaît le plus dans le tableau des lettres.
- Triez le tableau qui contient les 100 lettres aléatoires par ordre alphabétique. La sortie pourrait ressembler à ceci :

Les lettres minuscules sont :

```
e y l s r i b k j v j h a b z n w b t v  
s c c k r d w a m p w v u n q a m p l o  
a z g d e g f i n d x m z o u l o z j v  
h w i w n t g x w c d o t x h y v z y z  
q e a m f w p g u q t r e n n w f c r f
```

Les occurrences de chaque lettre sont :

```
a a a a b b b c c c c d d d d e e e e f f f f g g g g h h h i i i j j  
k k l l m m m m n n n n n o o o o p p p q q q r r r r s s t t t t  
u u u v v v v w w w w w w x x x y y z z z z z z
```

- Triez le tableau qui contient les 100 lettres aléatoires par les occurrences de la lettre dans l'ordre décroissant. Cela signifie que vous commencez par les lettres qui apparaissent le plus et terminez par les lettres qui apparaissent le moins. Ex:  
entrée : a b b c c c d e e e e  
sortie : e e e e c c c b b d a