

# prog sys

## Question 1

Il suffit d'utiliser la méthode `Math.random()` pour générer un nombre aléatoire dans une interval de 0 à la taille -1 du tableau.

## Question 2

On crée une classe `ThreadMatch` qui implémente `Runnable` ayant deux attributs pour stocker le nom des équipes.

Dans une boucle qui tourne 5 fois on va utiliser `Thread.sleep()` avec en paramètre le nombre de milliseconde allant de 0 à 10000 pour 10 seconde. Ensuite on affiche le nom de l'équipe et le commentaire.

## Question 3

Il faut ici créer 3 instance de `ThreadMatch` et les lancer 1 par 1 avec la méthode `start()`.

## Question 4

Ici à la suite de nos utilisations de `start()` pour lancer les threads on va aussi utiliser `join()`. On remarquera donc qu'on perd l'exécution en parallèle et donc tout l'intérêt des threads.

## Question 5

On commence par ajouter un nouveau paramètre à notre classe `ThreadMatch` ce paramètre sera un sémaphore.

On va ensuite vérifier à chaque commentaire généré s'il exprime un but, on va donc utiliser la méthode `startsWith("But")` pour vérifier s'il y a un but.

S'il y a un but on va "ajouter un jeton" à notre sémaphore passé un paramètre avec la méthode `.release(1)`, ceci aura pour effet de débloquer un code en attente de ce jeton.

Justement on va créer ce code dans notre classe de test. Pour cela on va juste faire une boucle infini qui va essayer de récupérer un jeton dans le sémaphore créé en amont et passé en paramètre de nos threads, on fait cela avec la méthode `.acquire(1)`.

On peut ensuite afficher "BUT !!!!!!!".

## Question 6

Ici on va encore utiliser les sémaphores, on va en passer un nouveau en paramètre de nos `ThreadMatches`.

Dès qu'une mi temps sera passé on va ajouter 1 jeton à ce sémaphore. Dans la classe de test on va essayer de lui en retirer 3 d'un coup (ce qui signifiera que les 3 mi temps sont terminés).

On va ensuite créer un autre sémaphore pour relancer les deuxièmes mi temps qu'on va initialiser à 0, on va également le passer en paramètres de `no Threads`.

Dès que le thread principale sera débloqué à la fin des 3 mi temps, on va ajouter 3 à la valeur de notre deuxième thread pour débloquer les autres threads.