

TP et labo de la semaine 5

- 1) Nous sommes intéressés à vérifier à quel point la fonction `Math.random()` donne des nombres aléatoires. Pour cela, nous voulons simuler une série de tirs à pile-ou-face. Nous voulons compter le nombre de fois qu'il faut tirer pour que 5000 des tirs aient donné pile. En principe le nombre de tirs devrait être proche de 10000 si `Math.random` est aléatoire. Codez un programme qui effectue cette simulation dans un premier temps avec une boucle `while`, et ensuite avec une boucle `for`.
Note : pour tirer à pile-ou-face on peut se servir du test `Math.random() < 0.5` . Un résultat `"true"` signifie "pile", et un résultat `"false"` signifie "face".
- 2) Codez un programme qui, étant donné une variable `n` qui contient un entier ≥ 0 , calcule et affiche la somme des `n` premiers nombres impairs (par exemple, pour `n=4` : `1 + 3 + 5 + 7`). Essayez d'exécuter votre programme avec codeBoot avec différentes valeurs de `n`. En mots expliquez la relation entre la somme obtenue et la valeur de `n`. Trouvez une optimisation (meilleur codage) de votre programme pour qu'il calcule la même valeur en moins de pas d'exécution.
- 3) Dans un programme nous voulons obtenir de l'utilisateur un nombre positif. Si l'utilisateur entre un nombre invalide (`<= 0` ou bien du texte non-numérique) nous voulons continuer à lui demander un nombre positif jusqu'à ce qu'il entre un nombre positif. Codez cette partie de programme.
- 4) Utilisez la réponse de l'exercice #3 pour coder un programme qui affichera la somme de tous les nombres positifs qui ont été entrés par l'utilisateur. L'utilisateur indiquera qu'il a fini d'entrer des nombres en entrant le nombre `-1`. Le programme doit indiquer l'index du nombre qui est demandé (i.e. le premier, le second, etc) avec le message "Entrez le nombre positif #1", "Entrez le nombre positif #2", etc.
- 5) Modifiez le programme de l'exercice #4 pour qu'il affiche la moyenne des nombres positifs entrés par l'utilisateur.
- 6) Affichez un damier de taille `N x N`. Expliquez d'abord votre algorithme en mots, puis exprimez-le en JavaScript. (Source: Eloquent JavaScript)

Un damier 8x8

```
# # # #
# # # #
# # # #
# # # #
# # # #
# # # #
# # # #
# # # #
```