

But de la séance : utiliser des boucles de toute nature, ainsi que des tableaux 1D pour mémoriser et transmettre des données à l'intérieur d'un programme.

Les tableaux sont des variables de type référence, ie la variable contient l'adresse d'une zone mémoire. Conséquence de cela, si on modifie, dans une méthode, le contenu d'un tableau passé en paramètres, ce contenu sera aussi modifié pour l'appelant. C'est à la fois bien (cela peut simplifier l'écriture du code) et pas bien (coté sûreté de codage cela peut amener des comportements non désirés si on n'y prend pas garde). Il vaudrait mieux systématiquement retourner un tableau « résultat » en fin de méthode. Mais cela obligerait à dupliquer le tableau dans la méthode afin d'être sûr qu'il n'y ait pas d'effet de bord entre une méthode manipulant un tableau et d'autres méthodes (c'est en particulier le cas dans un environnement concurrent utilisant des threads).

Les exercices seront traités dans l'ordre, l'important n'est pas le nombre réalisé à la fin de la séance, mais votre compréhension et votre capacité à le refaire.

Table des matières

Exercice 1 – Traces d'algorithmes.....	2
Exercice 2 – Petits jeux.....	4
Exercice 3 – Tableaux 1 dimension.....	4

Exercice 1 - Traces d'algorithmes

On donne l'algorithme suivant :

Algorithme 1

```
debut
  s ← 0
  u ← 3
  n ← 2
  POUR i allant de 1 à 5 FAIRE
    n ← (-1) * n
    u ← u * i
    s ← s + n * u
  FIN
fin
```



Question 1

- a) Calculez la trace de cet algorithme
- b) À quelle(s) condition(s) fonctionne-t-il ?

On donne l'algorithme suivant :

Algorithme 2

```
debut
  n ← 4
  s ← 256
  POUR i allant de 1 à 3 FAIRE
    s ← RACINE(s) / n
  FIN
fin
```



Question 2

- a) Calculez la trace de cet algorithme
- b) À quelle(s) condition(s) fonctionne-t-il ?

On donne l'algorithme suivant :

Algorithme 3

```
debut
    s ← 0
    POUR i allant de 1 à 3 FAIRE
        POUR i allant de 1 à 2 FAIRE
            s ← s + 2 * i + i * j
        FIN
    FIN
fin
```



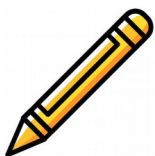
Question 3

- a) Calculez la trace de cet algorithme
- b) À quelle(s) condition(s) fonctionne-t-il ?

On donne l'algorithme suivant :

Algorithme 4

```
debut
    u ← 1
    n ← 0
    TANT QUE u ≤ 100 FAIRE
        u ← 2 * u * u
        n ← n + 1
    FIN
fin
```



Question 4

- a) Calculez la trace de cet algorithme
- b) Formulez un énoncé correspondant à ce programme

Exercice 2 - Petits jeux



Question 1

Écrire un programme qui génère un entier aléatoire n entre 10 et 100 inclus, qui demande à l'utilisateur d'entrer un entier d entre 3 et 7 inclus (on redemande la saisie tant qu'elle n'est pas valide), si n est divisible par d , l'utilisateur a gagné.

Question 2

Écrire une 2ème version où on demande ensuite à l'utilisateur s'il veut rejouer. Tant qu'il saisit 'o' il rejoue, à chaque fois on affiche les valeurs de n , d et "gagné" ou "perdu".

Question 3

Écrire un programme qui génère un entier aléatoire N compris entre 0 et 100, puis qui demande à l'utilisateur d'entrer un entier entre 0 et 100, si l'utilisateur n'a pas saisi N , on lui indique si le nombre cherché est plus grand ou plus petit, et on lui permet de recommencer jusqu'à ce qu'il trouve N . Lorsque l'utilisateur a trouvé N , on lui indique le nombre d'essais qu'il a fait, puis le message "pas mal" s'il a eu besoin de 5 essais ou moins, et le message "trop nul" s'il a eu besoin de plus de 5 essais.

Exercice 3 - Tableaux 1 dimension

Dans cet exercice, un programme principal a pour rôle d'appeler des méthodes de gestion d'un tableau. Il y aura donc deux classes : l'une comportant la méthode principale **main**, l'autre implémentera le problème à traiter.



Question 1

Écrire un programme qui remplit un tableau 1D de 20 cases par des nombres aléatoires compris entre -10 et +10, qui les affiche, qui en calcule la moyenne, le min et le max (et qui les affiche). Relancer le programme pour un tableau de 1 000 nombres (on désactivera l'affichage du tableau), 1 000 000 de nombres... Que valent les valeurs extrêmes et la moyenne ?

Question 2

Rajouter au programme précédent une méthode qui permet à l'utilisateur de saisir un nombre qui sera inséré à sa place (à condition que ce nombre soit compris entre -10 et +10). Après insertion, le tableau sera de nouveau affiché pour vérifier que le comportement est correct.