

Consolidation des acquis : Les tableaux et les structures

Exercice 0 : Utilisation des tableaux

- 1) Créer un tableau de 10 entiers.
- 2) Initialiser le tableaux de sorte que la case i contient l'entier i
- 3) Afficher le tableau

Exercice 1 : Crible d'Ératostène

Le crible d'Ératostène est un procédé qui permet de trouver tous les nombres premiers inférieurs à un certain entier naturel donné N .

L'algorithme procède par élimination : il s'agit de supprimer d'une table tous les multiples des entiers de 2 à N .

On commence par les multiples de 2, puis à chaque fois on raye les multiples du plus petit entier restant jusqu'à ce que le carré de celui-ci soit supérieur au plus grand entier de la liste.

On peut s'arrêter lorsque le carré du plus petit entier est inférieur au plus grand entier, car dans ce cas, s'il existait des non-premiers, ils auraient déjà été rayés précédemment.

À la fin du processus, tous les entiers qui n'ont pas été rayés sont les nombres premiers inférieurs à N .

Implémentez en Java la fonction `int nombre_permiers(int n)` qui compte le nombre de nombre premiers plus petit que n à l'aide du crible d'Ératostène.

Exercice 2 : Les ensembles d'entiers

Il est possible de coder les ensembles d'entiers à l'aide des tableaux.

Soi E un ensemble, et m son élément maximal. Alors il est possible de coder E à l'aide d'un tableau T de booléen à condition que son nombre de case soit strictement plus grand que m .

Pour cela, il suffit de considérer chaque case située à la position i du tableau comme étant le booléen qui code la présence de l'élément i dans E .

Par exemple, l'ensemble $\{0, 3, 5, 6\}$, se code à l'aide, du tableau suivant:

0	1	2	3	4	5	6	7	8
True	False	False	True	False	True	True	False	False

- 1) Implémenter la fonction `void ajouter(int E[], int i)` qui ajoute l'élément i à l'ensemble E .
- 2) Implémenter la fonction `boolean supprimer(int E[], int i)` qui supprimer l'entier i de l'élément E .
- 3) Implémenter la fonction `boolean appartient(int E[], int i)` qui dit si un entier i appartient à l'ensemble E .
- 4) Implémenter la fonction `int[] union(int E[], int F[])` qui renvoi un nouveau tableau étant l'union des deux tableaux E et F .

- 5) Implémenter la fonction `int[] intersection(int E[], int F[])` qui renvoi un nouveau tableau étant l'intersection de E et de F .
- 6) Implémenter la fonction `int[] moins(int E[], int F[])` qui renvoi un nouveau tableau étant les éléments de E privés de ceux de F .

Exercice 4 : Classe d'étudiants

Nous souhaitons implémenter une classe d'étudiants. Chaque étudiant possède un nom, et une note.

- 1) Implémenter une fonction qui permet d'ajouter un nouveau étudiant avec une note.
- 2) Implémenter une fonction qui calcule la moyenne des étudiants.
- 3) Implémenter une fonction qui donne le nom et la note du meilleur élève.