

## Présentation

Le but de ce TP est de manipuler les notions vues dans les cours et TP précédents, en particulier les tableaux, tests et boucles.

### 1 Définition

Un nombre entier  $p \in \mathbb{N}$  est premier ssi  $p > 1$  et  $p$  n'est divisible que par 1 et par lui-même. On notera  $P = \{2, 3, 5, 7, \dots\}$  l'ensemble des nombres premiers. De plus, pour tout  $i \in \mathbb{N}$ ,  $p_i$  sera le  $i^{\text{ème}}$  nombre premier, de sorte que  $p_0 = 2$ ,  $p_1 = 3$ ,  $p_2 = 5$ ,  $p_3 = 7$ , etc.

### 2 Approche naïve

#### Exercice 1

Écrivez un programme qui demande à l'utilisateur d'entrer un nombre entier naturel, et qui teste si ce nombre est premier. Si le nombre n'est pas premier, le programme doit afficher le plus petit diviseur. Le programme devra tester tous les diviseurs possibles en partant de 2.

*exemple 1 :*

```
Entrez le nombre naturel a tester : 17
17 est un nombre premier.
```

*exemple 2 :*

```
Entrez le nombre naturel a tester : 12
12 n'est pas un nombre premier : il est divisible par 2.
```

**Rappel :** l'opérateur permettant d'obtenir le reste (modulo) de la division entière est %.

#### Exercice 2

Dans le main, faites une copie du code source de votre programme précédent, et modifiez-la de manière à afficher tous les nombres premiers compris entre 1 et la valeur entrée par l'utilisateur (en incluant cette valeur).

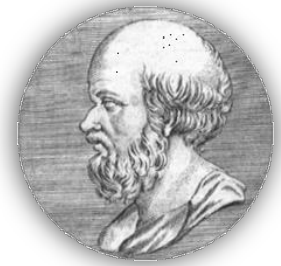
```
Entrez un entier naturel (>2) : 9
1 est un nombre premier.
2 est un nombre premier.
3 est un nombre premier.
5 est un nombre premier.
7 est un nombre premier.
```

#### Exercice 3

Faites une copie de votre programme précédent, et modifiez-la de manière à stocker tous les nombres premiers dans un tableau nommé `premier`, avant de les afficher. L'élément  $i$  du tableau doit contenir le  $i^{\text{ème}}$  nombre premier : `premier[0]` contient 2, `premier[1]` contient 3, `premier[2]` contient 5, `premier[3]` contient 7, etc.

### 3 Crible d'Ératosthène

L'inconvénient de l'algorithme naïf est qu'il effectue de nombreuses divisions inutiles. Par exemple, il peut tester la divisibilité par 6 d'un entier qui n'est divisible ni par 3, ni par 2. [Ératosthène](#) était un scientifique de l'antiquité, qui a proposé un algorithme appelé [Crible d'Ératosthène](#), qui permet d'éviter ces calculs inutiles.



#### Exercice 4

L'algorithme du Crible d'Ératosthène pour déterminer les nombres premiers compris entre 2 et  $N$  est le suivant :

- On considère un tableau contenant tous les entiers de 2 à  $N$ .
- On supprime tous les multiples de 2 (sauf 2).
- Parmi les nombres restants, on supprime tous les multiples de 3 (sauf 3).
- On répète ce traitement jusqu'à ce qu'il n'y ait plus de nombre à traiter.
- Les nombres restants à la fin du traitement sont les nombres premiers recherchés.

Appliquez manuellement cet algorithme pour trouver les nombres premiers compris entre 1 et 30.

#### Exercice 5

Il est peu pratique d'implémenter directement cet algorithme ici, en raison de la difficulté à supprimer des éléments dans un tableau. On se propose plutôt d'adapter notre programme de façon à exploiter le principe d'Ératosthène :

- Soit le tableau `premier`, utilisé pour stocker les nombres premiers ;
- On considère chaque entier de 2 jusqu'à  $N$  ;
- Pour chaque entier  $n$ , on teste s'il est premier en le divisant par les nombres premiers inférieurs à  $n$  (qui sont contenus dans le tableau `premier`) ;
- Si  $n$  est premier, on le rajoute dans le tableau. Sinon, on ne fait rien ;
- On passe à  $n + 1$ .

Faites une copie du programme précédent et modifiez-la pour y intégrer cette modification.

### 4 Amélioration

#### Exercice 6

L'implémentation de cet algorithme nécessite de déclarer un tableau de taille  $N$ , ce qui requiert une grande quantité de mémoire (si  $N$  est grand). Pourtant, on identifie forcément moins de  $N$  nombres premiers, donc une partie de ce tableau est inutilisée. Il est possible d'utiliser un tableau plus petit en utilisant la propriété suivante :

$$P_1: n \in P \Leftrightarrow \left( \forall p \in P \cap \left[ 2; \frac{n}{2} \right], p \text{ ne divise pas } n \right)$$

Prouvez la propriété  $P_1$ .

#### Exercice 7

Faites une copie de votre dernier programme, et modifiez-la de manière à intégrer cette amélioration permettant d'utiliser un tableau de taille  $N/2$  au lieu de  $N$ .