

# Crible d'Eratosthène [th05] - Examen

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  Version 22 mai 2018

## Table des matières

<b>1</b>	<b>Crible d'Eratosthène / pgeratos</b>	<b>2</b>
1.1	Algorithme du Crible ( <b>5 points</b> )	2
1.2	Liste des nombres premiers ( <b>4 points</b> )	4
1.3	Programme ( <b>1 point</b> )	4

## C++ - Crible d'Eratosthène (Examen)



### Objectif

Cet exercice détermine tous les nombres premiers inférieurs à un entier  $n$  par la méthode du crible d'ERATOSTHÈNE.

...(énoncé page suivante)...

# 1 Crible d’Eratosthène / pgeratos

## 1.1 Algorithme du Crible (5 points)



### Définition

Un **entier** est dit **premier** s’il possède exactement deux diviseurs : 1 et lui-même appelés diviseurs triviaux. Sinon il est dit **composite**.



### Propriété

Le **crible d’Eratosthène**<sup>1</sup> permet de connaître en une seule fois un grand nombre d’entiers naturels premiers consécutifs et pas trop grands (par exemple inférieurs à un milliard).



(1 point) Définissez la constante `CMAx=100000` (nombre maximum de valeurs d’un crible) puis le type `Crible` comme étant un tableau de `CMAx` booléens.



(1 point) Écrivez une procédure `initialiserCrible(c,n)` qui initialise à `Vrai` les éléments d’un `Crible c[..n]`. Par défaut tous les nombres sont premiers. Marquez à `Faux`, les entiers 0 et 1.



(1 point) Écrivez une procédure `eliminerCrible(c,n,k)` qui marque à `Faux` tous les **multiples successifs** de `k` (entier), à savoir  $2k, 3k, \dots$ , dans un `Crible c[..n]`.



(1 point) Écrivez une fonction `suivantImpair(c,n,k)` qui recherche et renvoie le suivant **impair** non marqué de `k` (entier impair) dans un `Crible c[..n]`, et qui renvoie `-1` s’il n’existe pas (ce qui marquera la fin de la recherche).

---

1. Mathématicien et philosophe, connu pour ses travaux en arithmétique et en géométrie, ERATOSTHÈNE vécut au III<sup>e</sup> siècle avant J.C. à Alexandrie.

**Écrivez le code sur cette partie...**



**(1 point)** Écrivez une procédure `eratosCrible(c,n)` qui calcule les nombres premiers compris entre 1 et `n` dans un `Crible c`. La procédure comporte trois parties :

- L’initialisation par appel de la procédure `initialiserCrible`.
- L’élimination (procédure `eliminerCrible`) de tous les multiples de 2 (2 est le plus petit nombre premier).
- L’élimination de tous les multiples successifs de chaque entier **impair** en commençant par le plus petit 3.

## 1.2 Liste des nombres premiers (4 points)



**(1 point)** Définissez la constante `TMAX=1000` (nombre maximum de valeurs dans une liste), éventuellement le type `ITableau` comme étant un tableau d’entiers d’au plus `TMAX` entiers, puis le type `IListe` comme étant une structure contenant :

- Un `ITableau` contenant les valeurs.
- Un entier `taille` du nombre d’éléments effectifs dans le `ITableau`.



**(0.5 point)** Écrivez une procédure `initialiserListe(lt)` qui initialise une `IListe lt` à la liste vide (aucun élément, à savoir sa taille est nulle).



**(1 point)** Écrivez une procédure `ajouterElement(lt,valeur)` qui ajoute une valeur `valeur` (entier) dans une `IListe lt` (à la suite de ceux déjà présents).



**(1 point)** Écrivez une procédure `creerListeNP(n,lt)` qui crée la liste des entiers premiers inférieurs ou égaux à `n` dans une `IListe lt`.



**(0.5 point)** Écrivez une procédure `afficherListe(lt)` qui affiche les valeurs du tableau d’une `IListe lt`.

## 1.3 Programme (1 point)



**(1 point)** Écrivez un programme qui saisit un entier (supposé positif et inférieur à `CMAX`) puis calcule et affiche la liste des nombres premiers qui sont inférieurs ou égaux à cet entier.



Testez. Exemple d’exécution :

```
Entier dans [1..99999]? 120
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
101 103 107 109 113
```

**Écrivez le code sur cette partie...**