

EXERCICES

LES VARIABLES
(PART 2)

VERSION

1.0

AUTEURS

ADRIEN GODOY

Table des matières

Exercice 1 : les nombres premiers.....	3
1.1 - Q1 : Trouvez - à la main - tous les nombres premiers inférieurs à 30. ...	3
1.2 - Q2 : Proposez une fonction qui permet de calculer tous les nombres premiers inférieurs à un nombre donné N.	3
1.3 - Q3 : Le crible d'Eratosthène	4
Exercice 2 : Les nombres premiers jumeaux.....	5

1 - Exercice 1 : les nombres premiers

1.1 - Q1 : Trouvez – à la main - tous les nombres premiers inférieurs à 30.

en partant de 0, les nombres premiers jusqu'à trente sont :
2, 3, 5, 7, 11, 13, 17, 19, 23 et 29

bonus :

- ScienceEtonnante : Les nombres premiers
<https://www.youtube.com/watch?v=R37JHiA-HOg>
- Science4All : Les nombres premiers sont-ils (presque) aléatoires ?
<https://www.youtube.com/watch?v=kt8Uu37RHbo>

1.2 - Q2 : Proposez une fonction qui permet de calculer tous les nombres premiers inférieurs à un nombre donné N.

```
1  #include <stdio.h>
2  #include <stdbool.h>
3
4  void nbPremiers();
5
6  int main()
7  {
8      nbPremiers();
9      return 0;
10 }
11
12 void nbPremiers()
13 {
14     int nombremax;
15     int i;
16     int j;
17     bool divisible = false;
18
19     printf("\nJusqu'a quel nombre voulez-vous enumerer les nombres premiers ?\n");
20     scanf("%d", &nombremax);
21     getchar();
22     printf("\nnombres premiers :\n");
23
24     // est nombre est dit premier s'il n'est divisible que par 1 et lui-memeement
25     // autrement dit, si, si son modulo par un nombre different de 1 par tout autre
26     // nombre inferieur est different de 0
27     // d'abord, dans la premiere boucle for, on parcourt les entiers jusqu'au nombremax choisi
28     for (i = 2; i <= nombremax; i++)
29     {
30         divisible = false;
31
32         // on verifie le modulo de tous les nombres inferieurs
33         for (j = 2; j < i; j++)
34         {
35             if (i%j == 0)
36                 divisible = true;
37         }
38         if (divisible == false)
39             printf("%d\n", i);
40     }
41 }
```

remarque : on pourrait optimiser cet algorithme en créant un nouvel int qui serait la partie entière de la racine carrée de nombremax, puis en testant la division de nombremax par 2, et ensuite en ne testant que les nombres impairs jusqu'à la racine carrée de nombremax (en partant de 3 et en incrémentant de deux le compteur j).

1.3 - Q3 : Le crible d'Eratosthène

Tiens selon vous à quelle valeur de i (à comparer avec le nombre N) s'arrête le traitement ?

La partie entière de la racine carrée de N

```

63 void eratosthene()
64 {
65     int i;
66     int j = 2;
67     bool tab[nombremax+1]; // nombremax est une globale
68
69     // etape 0 : on passe toutes les valeurs du tableau a true
70     for (i = 2; i <= nombremax; i++)
71         tab[i] = true;
72
73     // etape 1 : 0 et 1 ne sont pas des nombres premiers
74     tab[0] = false;
75     tab[1] = false;
76
77
78     // etape 2 : parcourt les indices du tableau a partir de 2
79     for (i = 2; i <= floor(sqrt(nombremax)); i++)
80     {
81         // si le nombre n'est pas deja un multiple d'un nombre inferieur,
82         // on calcule tous ses multiples et on passe les indices correspondants a false
83
84         for (j = 2; j <= nombremax; j++)
85         {
86
87             if (tab[i] == true && i*j <= nombremax)
88             {
89                 tab[i*j] = false;
90             }
91         }
92     }
93
94     // etape 3 on affiche les indices don la valeur est restee true
95     for (i = 2; i <= nombremax; i++)
96     {
97         if (tab[i] == true)
98             printf("\n%d", i);
99     }
100
101     printf("\n");
102
103     jumeaux(tab, nombremax);
104
105 }
106

```

Exercice 2 : Les nombres premiers jumeaux

```
107 void jumeaux(bool *tab, int nombremax)
108 {
109     int i;
110     for (i = 2; i <= nombremax - 2; i++)
111     {
112         if (tab[i] == true && tab[i+2])
113             printf("\n%d et %d sont jumeaux", i, i+2);
114     }
115     printf("\n");
116 }
```