Exercice 2.3 - Diviseurs stricts

1. Méthode la plus évidente

La méthode la plus évidente pour identifier et sommer tous les diviseurs stricts d'un entier n et de parcourir l'ensemble des nombres compris entre 1 et $n-1$ et de vérfier s'ils divisent n.

Algorithm 1 Somme des diviseurs stricts (1)

```
Vars
    n : int                      ▷ Entier à analyser
    s : int                      ▷ Somme de ses diviseurs stricts
    d : int                      ▷ Diviseur potentiel en cours de vérification
EndVars

Begin
    print "n ?"
    get n

    s ← 0
    d ← 1
    while d < n do
        if n mod d = 0 then
            s += d
        end if
        d += 1
    end while
    print "S = " + s
End
```

Le nombre d'opérations effectuées dans cet algorithme est de l'ordre de grandeur de n.

2. Méthode plus efficace

L'algorithme précédent peut être amélioré en remarquant que pour tout diviseur d identifié, n/d est un diviseur ; on peut donc chercher les diviseurs d de 1 à $\sqrt{n}$ et ajouter d et n/d en prenant néanmoins garde, le cas échéant de n'ajouter q'une fois $\sqrt{n}$.

L'ordre de grandeur du nouvelle algorithme est donc de $n^{1/2}$ ce qui est beaucoup plus efficace que le précédent pour les grands nombres.

Algorithm 2 Somme des diviseurs stricts (2)

```
Vars
    n : int                                   ▷ Entier à analyser
    s : int                                   ▷ Somme de ses diviseurs stricts
    d : int                                   ▷ Diviseur potentiel en cours de vérification
EndVars

Begin
    print "n ?"
    get n

    s ← 0
    if n > 1 then
        s ← 1
        d ← 2
        while d < √n do
            if n mod d = 0 then
                s += d
                s += n div d
            end if
            d += 1
        end while
        if (n mod d = 0) AND (d = n div d) then      ▷ d est la racine carrée de n
            s += d
        end if
    end if
    print "S = " + s
End
```