

**Écriture de procédures en MAPLE.**

Voici une liste non exhaustive de fonctions Maple à utiliser : **ithprime**, **divisors**, **tau**, **type**, **convert**, **proc**, **map**, **union**,...

Il s'agit de créer des fonctions utilisant des boucles **for** diverses et variées.

1. créer une fonction **Somme** qui a pour paramètre d'entrée un entier  $n$  positif et en sortie donne la somme des entiers de 1 à  $n$ . Exemple :

```
> Somme(10);
```

55

2. créer une fonction **somprem** qui a pour paramètre d'entrée un entier  $n$  et calcule la somme  $n$  premiers nombres premiers. Exemple :

```
> somprem(4);
```

17

3. créer une fonction **parfait** qui a pour paramètre d'entrée un entier  $n$  et retourne la liste des nombres parfaits compris entre 1 et  $n$  (*un nombre parfait est un nombre qui est égal à la moitié de la somme de ses diviseurs*).

```
> parfait(1000);
```

[6, 28, 496]

4. créer une fonction **plouton** qui a pour paramètre d'entrée un entier  $n$  et retourne la liste des  $n$  premiers nombres ploutons (*un nombre plouton est un nombre qui a strictement plus de diviseurs que ses prédécesseurs*).

```
> plouton(20);
```

[1, 2, 4, 6, 12, 24, 36, 48, 60, 120, 180, 240, 360, 720, 840, 1260, 1680, 2520, 5040, 7560]

5. Écrire une fonction **amiable** qui trouve tous les couples de nombres amiables compris entre 1 et  $n$  (un couple d'entier  $(a, b)$  est dit amiable si  $a$  et  $b$  sont distincts et la somme des diviseurs de  $a$  moins  $a$  est égal à  $b$  et réciproquement. exemple : (220,284)). Exemple :

```
> amiable(3000);
```

{[1210, 1184], [220, 284], [1184, 1210], [284, 220], [2620, 2924], [2924, 2620]}

6. Écrire une fonction **palindrome** qui teste si un nombre est égal à son symétrique (exemple : 12321 et 457754 sont des nombres palindromes). Exemples :

```
> palindrome(12345698789654321);
```

*true*

```
> palindrome(123654078987654321);
```

*false*

7. Écrire à l'aide de deux boucles **for** imbriquées, un programme MAPLE qui donne les tables logiques de: a ou b, a et b,  $a \Rightarrow b$ ,  $a \Leftrightarrow b$ , sous une forme approchant la suivante :

a	b	a ou b	a et b	$a \Rightarrow b$	$a \Leftrightarrow b$
true	true	true	true	true	true
true	false	true	false	false	false
false	true	true	false	true	false
false	false	false	false	true	true

*indication* : on pourra se servir de la fonction **printf**.