

TP n°09.A : Les sous-algorithmes (Procédures)

Partie I: Algorithmes

Exercice N°01

Ecrire une procédure *lire (a)* qui permet de lire un nombre entier positif a .

Exercice N°02

Ecrire une procédure *trier (p,g)* qui permet de trier deux nombres entiers p et g .

Exercice N°03

Ecrire une procédure *divisible (n,d)* qui permet de savoir si l'entier n est divisible par d ou non. On pourra utiliser un nouveau paramètre nommé r afin de renvoyer le résultat *divisible (n,d,r)*.

Exercice N°04

Ecrire une procédure *parfait (x)* qui nous permet de dire si un nombre est parfait ou non, et qui appelle la procédure Divisible. On appelle un nombre parfait un nombre égal à la somme de ses diviseurs propres (un diviseur propre est un diviseur autre que le nombre lui-même). Ex: $6 = 1 + 2 + 3$ et: $28 = 1 + 2 + 4 + 7 + 14$

Partie II: Programmation en C

Ecrire un programme en langage C qui affiche tous les nombres parfaits compris entre deux valeurs entières positives lues x et y , en utilisant les procédures précédentes.

Exécutez le programme en donnant:

- 1) $x=1$ et $y=30$
- 2) $x=500$ et $y=450$
- 3) $x=9000$ et $y=8000$