



UNIVERSITE CHEIKH ANTA DIOP DE DAKAR
FACULTE DES SCIENCES ET TECHNIQUES
LABORATOIRE D'ALGEBRE, DE CRYPTOLOGIE, DE GEOMETRIE ALGEBRIQUE ET
APPLICATIONS
(LACGAA)

TP4 ALGORITHMIQUE

Exercice 1 :

Écrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants.

Par exemple, si l'utilisateur entre le nombre 35, le programme affichera les nombres de 36 à 45.

Résoudre ce problème avec trois boucles :

- **Pour.**
- **Tant Que.**
- **Répéter.**

Exercice 2 :

Écrire un algorithme qui permet d'afficher les entiers de 75 à 100 en ordre inverse.

Exercice 3 :

1. Écrire un algorithme qui demande à l'utilisateur un nombre compris entre 0 et 9(inclus) jusqu'à ce que la réponse convienne.
2. Écrire un algorithme qui force l'utilisateur à donner un nombre strictement positif.

Exercice 4 :

Écrire un algorithme qui demande un nombre entier strictement positif et calcule la somme des entiers jusqu'à ce nombre(à partir de 1).

Exemple : **L'utilisateur donne 5 et l'algorithme fait $1+2+3+4+5$ et affiche 15.**

NB : on souhaite afficher uniquement le résultat, pas la décomposition du calcul.

Exercice 5 :

Écrire un algorithme qui affiche tous les entiers pairs de 1 à 24.

Exercice 6 :

Écrire un algorithme qui demande un entier $n > 0$ et donne comme résultat :

$$som = \begin{cases} 1 + 3 + 5 + 7 + \dots + n, & \text{si } n \text{ est impair} \\ 2 + 4 + 6 + 8 + \dots + n, & \text{si } n \text{ est pair} \end{cases}$$

NB : n ne doit pas être négatif ou nul.

L'algorithme doit donner un seul résultat selon la valeur de n .

Exercice 7 :

Écrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « **Plus petit !** », et inversement, « **Plus grand !** » si le nombre est inférieur à 10.

Exercice 8 :

Écrire un algorithme qui détermine si un nombre (**positif et strictement supérieur à 1**) saisi au clavier est parfait ou pas.

Un nombre est parfait s'il est égal à la somme de ses diviseurs propres.

Exemple : 6 est parfait car en effet 1, 2 et 3 sont les diviseurs propres de 6 et leur somme est $1 + 2 + 3 = 6$.

Exercice 9 :

Écrire un algorithme qui détermine si un nombre (**positif et strictement supérieur à 1**) saisi au clavier est premier ou pas.

Un nombre est premier s'il n'est divisible que par 1 et par lui-même.

Exercice 10 :

Écrire un algorithme permettant d'afficher les diviseurs d'un entier positif saisi au clavier.

Attention : si ce nombre est nul, tous les entiers le divisent !

Exercice 11 :

Écrire un algorithme qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres.

Exercice 12 :

Proposer un algorithme qui demande à l'utilisateur un entier strictement positif n et affiche son écriture en base binaire (base 2).

Bonne Chance