



Module : Programmation et structure de données

Semestre : 02

T.D. N° 03

Les Fonctions et les Procédures

Exercice n° :01

Ecrire une fonction qui calcule le **PGCD** de deux entiers **A** et **B** strictement positifs.

Exercice n° :02

Ecrire une fonction qui affiche si un nombre est premier ou non.

Exercice n° :03

Ecrire un algorithme (utiliser une fonction) affichant tous les nombres parfaits inférieurs à 10000. Sachant qu'un nombre entier positif (N) est parfait s'il est égal à la somme de ses diviseurs ($<N$).

Exemples : **6** qui est égal à **1 + 2 + 3** et **28** qui est égal à **1 + 2 + 4 + 7 + 14** — sont des nombres parfaits.

Exercice n° :04

Ecrire une procédure qui permet de résoudre une équation du second degré dans **R**.

Exercice n° :05

Ecrire une fonction qui permet de calculer la multiplication de deux nombres **A** et **B** entiers en utilisant l'addition.

Exercice n° :06

Ecrire une fonction ou une procédure qui permet d'avoir un nombre entier positif et afficher son image miroir. **Exemple** le nombre est **3524**, on doit afficher **4253**.

Exercice n° :07

Ecrire une fonction qui retourne la somme des éléments pairs du tableau **T** à **n** éléments.

Exercice n° :08

Écrire une fonction ou une procédure qui calcule le maximum **MAX** entre trois variables **A**, **B** et **C** de type entier.

Exercice n° :09

Écrire une fonction qui calcule la plus grande valeur (**MAX**) et leur position (**POSMAX**) de ce tableau **T** de taille **n**.

Exercice n° :10

Écrire une procédure qui permet de **trier** un tableau **T** de Taille **n** par ordre croissant.

Exercice n° :11

Écrire un algorithme affichant tous les nombres inférieurs à 500 égaux à la somme des cubes de leurs chiffres. On utilisera une fonction UNITE, et une fonction CUBE.

Exemple : $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27$.

Exercice n° :12

Écrire une fonction **BIN** permettant de convertir un entier positif du **décimal** au **binaire**.

Exercice n° :13

Écrire une fonction RACINE2 qui calcule la racine carrée d'un nombre positif en utilisant la formule suivante : $X_{i+1} = 1/2 * (X_i + a/X_i)$

Où $\sqrt{a} = X_{i+1}$ avec une précision **ER** = $|X_{i+1} - X_i|$ (ex. **ER**= 10^{-3}).

Exercice n° :14

Écrire un algorithme qui permet d'éclater un vecteur **T** de taille **N** ($N \leq 250$) entiers supposés positifs en deux vecteurs **T1** et **T2** contenant respectivement les nombres **pairs** et **impairs** de **T**.

Exercice n° :15

Soit une matrice carrée **A(N, N)** d'entiers ($N \leq 25$). Écrire un algorithme permet de :

- 1- Calculer la trace de la matrice A. (La trace est la somme des éléments de la diagonale principale).
- 2- Déterminer le maximum et sa position, des valeurs des deux diagonales (**principale** et **secondaire**).

Exercice n° :16

Écrire une fonction qui détermine si un mot est un **palindrome**. (Un mot palindrome se lit de gauche à droite et de droite à gauche (ex : **RADAR, ELLE, ICI**)).

Exercice n° :17

Écrire un algorithme qui calcule le nombre de caractères, de mots et de phrases dans un texte. Les mots sont séparés par des espaces et les phrases séparées par un point. (Sans compter les séparateurs : espace et point).

Exercice n° :18

Écrire un algorithme qui lit deux mots et qui détermine s'ils sont anagrammes. Sachant qu'un mot est dit anagramme d'un autre mot s'ils utilisent (sont formés par) les mêmes lettres.

Exemples :

CHIEN anagramme de **CHINE, NICHE,**

AIMER anagramme de **MAIRE, MARIE, RAMIE,**

GELER n'est pas anagramme d' **ALGER, ...**