TD n° 2 : Modularité

Exercice 1 :

Ecrire un algorithme qui calcule $C_n^p = \frac{n!}{p!}(n-p)!$, en utilisant une fonction qui détermine le factoriel d'un nombre entier naturel N.

Exercice 2 :

Ecrire un algorithme affichant tous les nombres (inférieurs à un entier naturel donné) qui sont égaux à la somme des cubes de leurs chiffres.

Utiliser une fonction CUBE pour le calcul du cube de chaque unité de chiffre et une fonction VERIF qui vérifie si un nombre est égal à la somme des cubes de ses chiffres.

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

Exercice 3 : Passage des paramètres (Par valeur / Par adresse)

Soit le programme suivant:

```
Algorithme Par
Var x, y, z, t: entiers

Procedure proc(a :entier, b :entier, var c : entier, d :entier)
Debut
  c ← a + b
  d ← a × b
Fin

Debut
Lire (x)
Lire (y)
proc(x, y, z, t)
Ecrire (z)
Ecrire (t)
FIN
```

1. Identifier les paramètres réels (Passage par valeur) et les paramètres formels (Passage par adresse).
2. Qu'affiche ce programme en supposant que l'utilisateur saisisse **2** dans **x** et **3** dans **y**? Modifier le code pour obtenir un résultat plus logique.

Exercice 4 : Effets de bord

Dérouler l'algorithme suivant en donnant les différentes valeurs des résultats attendus dans l'ordre d'exécution de l'algorithme

```
Algorithme Appel
Var R , V : entier

Fonction CALCUL (X : entier) : entier
Var R : entier
Debut
        R ← X + V
        V← R – 2
        CALCUL ← R + 2 * V
        Ecrire ( R , V )
Fin

Debut
V ← 5
R ← CALCUL(V) ; Ecrire ( R , V )
R ← CALCUL ( V ) ; Ecrire ( R , V )
R ← 10
V ← CALCUL ( R ) ; Ecrire ( R , V )
Fin
```

Exercice 5 :

Ecrire un algorithme qui fait appel à une fonction **BIN** permettant de convertir un entier positif du décimal au binaire.

Exercice 6 :

Ecrire une fonction ou procédure qui calcule la partie entière d'un nombre positif.

Exercice 7 :

Ecrire une fonction ou procédure qui calcule le PGCD de deux entiers strictement positifs

Exercice 8 :

Ecrire un algorithme qui fait appel à un sous-programme afin de calculer la somme suivante :

$$S = 1 - \frac{1}{X+1} + \frac{2}{X^2+2} - \frac{3}{X^3+3} + \cdots + \frac{N*(-1)^n}{X^n+n}$$

Exercice 9 :

Ecrire un sous-programme qui permet d'avoir en entrée un nombre entier positif et afficher son image miroir. Exemple le nombre est 72345 a comme image miroir : 54327.

Exercice 10 :

Écrire un algorithme qui lit une chaine de caractères et fait appel à un sous-programme afin d'afficher le miroir de chaque mot qui la compose, Ex : miroir ("hello") est "olleh".
NB : La longueur de la chaine de caractères ph est donnée par la fonction Longueur (phrase).

Exercice 11 :

Ecrire un sous-programme qui retourne Vrai si le caractère passé en paramètre est égal à 'o' ou 'O' (qui veut dire Oui), et Faux sinon.
2- Ecrire sous-programme qui permet d'afficher la table de multiplication de 1 à 9 d'un nombre entier positif.
3- En utilisant les sous-programmes précédents, écrire un algorithme permettant de saisir un nombre et d'afficher à l'utilisateur sa table de multiplication jusqu'à ce que la réponse introduite soit fausse.

Exercice 12 :
Déterminer le plus petit multiple commun ($PPCM$), de deux nombre entiers naturels A et B, en appliquant la formule :

$$PPCM(A,B) = \frac{A \times B}{PGCD(A,B)},$$
$$PGCD(A,0) = A \; et$$
$$PGCD(A,B) = PGCD(B, A \; mod \; B)$$