Série d'exercices N° 05

Les Procédures et Les Fonctions

Exercice 1 :

Écrire une fonction qui prend un nombre réel x en paramètre, puis retourne $f(x)$ tel que :

$$f(x) = \frac{\sqrt{x^2 - 2sin(x+4)}}{x^3}$$

```
Algorithme Exercice1

fonction f(x:réel) : réel
var y : réel
début
    y = racine(x^2 - 2*sin(x+4))/x^3
    retourner  y
Fin

Début
ecrire(f(7))
Fin
```

Exercice 2 :

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

```
Algorithme Exercice2
fonction multiplication(A: entier , B :entier ) : entier
var S,i : entier
debut
    S = 0
    pour i de 1 à B faire
        S = S + A
    FinPour
    retourner S
Fin
Début
Ecrire(multiplication(3,5))
Fin
```

Exercice 3 :

Écrire une procédure ValeurPair(M,N) qui prend deux paramètres M et N et d'afficher toutes les valeurs paires entre M et N si $M < N$

```
Algorithme Exercice3
procedure valeurPair(M:entier,N:entier)
var i : entier
```

```
début
    pour i de M à N faire
        Si i mod 2 = 0 alors
            Ecrire(i, ", ")
        FinSi
    FinPour
Fin
// programme principal : main
Début
    valeurPair(5,100)
Fin
```

Exercice 4 :

Écrire une fonction estPair(N) qui vérifie si un nombre N passé en paramètre est pair ou non

```
Algorithme Exercice4

fonction estPair(N:entier) : booléen
debut
    si N mod 2 = 0 alors
        retourner Vrai
    sinon
        retourner Faux
    FinSi
    retourner N mod 2 = 0
Fin
Début
Ecrire(estPair(8))
Fin
```

Exercice 5 :

Écrire une fonction max(a,b) qui retourne le maximum entre les deux paramètres a et b

Exercice 6 :

Écrire une fonction min(a,b) qui retourne le minimum entre les deux paramètres a et b

Exercice 7 :

Écrire une procédure swap(a,b) qui échange les valeur des deux paramètres a et b

```
Algorithme Exercice7
var x, y : entier
procedure swap(var a : entier,var  b : entier)
var c : entier
debut
    c = a
    a = b
    b = c
Fin
Début
x = 7
y = 5
ecrireln("x :", x, ", y : ", y)
swap(x,y)  // passage de paramètre e ->f : (valeur, adresse)
ecrireln("x :", x, ", y : ", y)
Fin
```

Exercice 8 :

Écrire une fonction partieEntiere(x) qui retourne la partie entière d'un nombre positif

```
Algorithme Exercice8

fonction partieEntiere(x: réel) : entier
var i : entier
debut
    i = 1
    tantque i <= x faire
        i=i+1
    FinTantQue
    retourner i -1
Fin
Début
Ecrire(partieEntiere(9.7))
Fin
```

Exercice 9 :

Écrire une procédure TableMultiplication(N) qui affiche le tableau de multiplication d'un entier positif N passé en paramètre

```
Algorithme Exercice9
procedure tableMultiplication(N :entier)
var i : entier
debut
    pour i de 1 à 10 faire
        Ecrireln(N, " x ", i , " = " , N*i)
    FinPour
Fin
Début
    tableMultiplication(3)
Fin
```

Exercice 10 :

Écrire une fonction fact(N) qui calcule la factoriel d'un nombre N positif

```
Algorithme Exercice10
fonction fact(N : entier) : entier
var p,i : entier
début
    p = 1
    pour i = N à 1 pas -1 faire
        p  = p*i
    FinPour
    retourner p
Fin
Début
Ecrire(fact(6))
Fin
```

Exercice 11 :

Écrire une fonction qui prend en paramètre un nombre entier positif et retourne son image miroir. Exemple le nombre est 3524, on doit afficher 4253.

```
Algorithme Exercice11

fonction miroir(N : entier) : entier
var inv : entier
début
inv = 0
tantQue N <> 0 faire
    inv = inv*10 + N mod 10
    N = N div 10
FintantQue
    retourner inv
Fin
Début
Ecrire(miroir(1234567))
Fin
```

Exercice 12 :

On appelle "palindrome" un mot ou une phrase qui se lit de la même façon à l'endroit comme à l'envers, sans tenir compte des espaces. Exemple : le mot "ABCBA" est un palindrome.
Écrire une fonction estPalindrome(ch) permettant de vérifier si une chaîne de caractères CH est un palindrome.

```
Algorithme Exercice12

fonction estPalindrome(ch : chaine):booléen
var i,j : entier
debut
    i = 0
    j = long(ch) -1
    tantQue i < j faire
        Si ch[i] <> ch[j] alors
            retourner  faux
        FinSi
        i = i + 1
        j = j - 1
    FinTantQue
    retourner Vrai
Fin
Début
    Ecrire(estPalindrome("ABCDBA"))
Fin
```

Exercice 13 :

Écrire une fonction qui détermine si un nombre est premier ou non (un nombre premier n'est divisible que par 1 et par lui-même)

```
Algorithme Exercice13
fonction estPremier(N : entier) : booléen
var i : entier
debut
    Si N=1 ou N=2 alors
        retourner Vrai
    FinSi
    pour i de 2 à N -1 faire
        Si N mod i = 0 alors
            retourner Faux
```

```
        FinSi
    FinPour
    retourner Vrai
Fin
Début
    ecrire(estPremier(6))
Fin
```

Exercice 14 :

Écrire une fonction qui détermine si un nombre N est parfait ou non (un nombre est parfait s'il est égale à la somme de ses diviseurs)

```
Algorithme Exercice14
fonction estParfait(N:entier):booléen
var i,S : entier
début
    S = 0
    pour i de 1 à N -1 faire
        Si N mod i = 0 alors
            S = S + i
        FinSi
    Finpour
     retourner N=S
Fin
Début
Ecrireln(estParfait(6))
Ecrireln(estParfait(28))
Ecrireln(estParfait(496))
Fin
```

Exercice 15 :

Écrire une fonction estBissextile(A) qui prend en paramètre une année A. poui vérifie si cette année est bissextile

```
Algorithme Exercice15
fonction estBissextile(A:entier):booléen
debut
    retourner A mod 1000 = 0 ou (A mod 4 = 0 et A mod 100 <> 0)
Fin
Début
ecrireln(estBissextile(2023))
ecrireln(estBissextile(2024))
ecrireln(estBissextile(2000))
ecrireln(estBissextile(1900))
Fin
```

Exercice 16 :

Écrire une fonction dateValide(J,M,A) qui prend en paramètres le jour J, le mois M et l'année A. Puis vérifie si la date est valide

```
Algorithme Exercice16
fonction estBissextile(A:entier):booléen
debut
    retourner A mod 1000 = 0 ou (A mod 4 = 0 et A mod 100 <> 0)
Fin
fonction dateValide(J : entier, M : entier , A : entier) : booléen
```

```
début
    Si J < 1 ou J > 31 ou M < 1 ou M > 12 alors
        retourner Faux
    sinonSi J=31 et (M=2 ou M=4 ou M=6 ou M=9 ou M=11) alors
        retourner Faux
    SinonSi J=30 et M=2 alors
        retourner Faux
    SinonSi J=29 et M=2 et estBissextile(A) = Faux alors
        retourner Faux
    sinon
        retourner Vrai
    FinSi
Fin
Début
    Ecrireln(dateValide(21,3,2023))
    Ecrireln(dateValide(40,3,2023))
    Ecrireln(dateValide(21,13,2023))
    Ecrireln(dateValide(31,6,2023))
    Ecrireln(dateValide(30,2,2023))
    Ecrireln(dateValide(29,2,2023))
    Ecrireln(dateValide(29,2,2024))
Fin
```

Exercice 17 :

Écrire un programme qui demande à l'utilisateur de saisir N réels stockés dans un tableau. Écrivez ensuite les fonctions suivantes :

- Une fonction permettant de remplir les éléments du tableau.
- Une fonction permettant de trier les éléments du tableau par ordre croissant.
- Une fonction permettant de trier les éléments du tableau par ordre décroissant.
- Une fonction permettant d'afficher les éléments du tableau.
- Tester toutes ces fonctions dans un programme principal

Exercice 18 :

Écrire une fonction qui calcule la distance de Hamming entre deux mots entrés par l'utilisateur. (La distance de Hamming entre deux mots de même longueur est le nombre d'endroits où les lettres sont différentes)

Exercice 19 :

Écrire une fonction nbr_occurence(ch,e) qui reçoit en arguments une chaine de caractères ch et un caractère e, la fonction retourne le nombre d'aoccurences du caractère e dans la chaine ch.

Exercice 20 :

Écrire une fonction pos_occurence(ch,e) qui reçoit en argument une chaine de caractères ch et un caractère e, la fonction retourne l'indice positif de la première occurrence du caractère e s'il existe dans la chaine ch. Sinon elle retourne -1

Exercice 21 :

Écrire une fonction cmp_chaine(ch1,ch2) qui reçoit en argument deux chaine de caractères ch1 et ch2, la fonction compare ch1 et ch2et retourne :

- 1 $si\ ch1 > ch2$

- 0 *si* $ch1 = ch2$
- -1 *si* $ch1 < ch2$

Exercice 22 :
Écrire une fonction chaine_chiffres(ch) qui reçoit en argument une chaine de caractères ch, la fonction retourne True si la chaine ch contient seulement des caractères chiffres sinon elle retourne False.

Exercice 23 :
Écrire une fonction supp_espace(ch) qui reçoit en argument une chaine de caractères ch, la fonction retourne la chaîne ch après avoir supprimé tous les caractères espaces s'ils existent au début de cette chaine.

Exercice 24 :
Écrire une fonction alphabetique(ch) qui reçoit en argument une chaine de caractères ch, la fonction retourne la chaîne ch après avoir supprimé tous les caractères qui ne sont pas des caractères alphabétiques dans cette chaine.

Exercice 25 :
Écrire une fonction qui convertit une chaîne de caractères, en minuscule puis en majuscule

Exercice 26 :
Écrire une fonction permettant de supprimer les espaces superflus (plusieurs espace successifs) dans une chaîne de caractères.

Exercice 27 :
Écrire une fonction qui returne la liste de tous les nombre premiers inférieurs à un nombre N passé en paramètre

Exercice 28 :
Écrire une fonction prend en paramètres deux mots ch1 et ch2 et qui retourne tous les caractères qui apparaissent dans les deux chaînes sans redondance.

Exercice 29 :
Écrire une fonction qui détermine si une adresse ip passé en paramètre et valide ou non :
exemples :

- "192.168.1.1" et valide
- "392.168.1.1" et n'est pas valide valide
- "192.168.1" et n'est pas valide valide

Exercice 30 :
Écrire une fonction qui retourne le masque de sous réseau pour un préfix réseau passé en paramètre
exemples : mask(16) returne "255.255.0.0"

Exercice 31 :
Écrire une fonction ou procédure qui calcule le PGCD de deux entiers strictement positifs.