

Atelier 06 : Les fonctions et procédures

Exercice 01 : Fonction Moyenne

En se basant sur l'algorithme, de l'exercice 01 de l'atelier 01, écrire le sous-algorithme de la fonction *Moyenne* qui renvoie la moyenne de deux entiers.

Ecrire un algorithme qui contient la déclaration de la fonction moyenne et des instructions qui appellent cette fonction.

Solution :

Sous-algorithme de la fonction Moyenne :

```
// Déclaration de la fonction Moyenne
Fonction Moyenne(X : entier, Y : entier) : réel
Début
    Retourner (X + Y) / 2
FinFonct
```

Algorithme :

Algorithme Fonction_Moyenne

Variables A, B : entiers

// Déclaration de la fonction Moyenne

Fonction Moyenne(X : entier, Y : entier) : réel

Début

 Retourner (X + Y) / 2

FinFonct

// Programme principal

Début

 // Saisie des données

 Ecrire ("Entrez la valeur de A:")

 Lire (A)

 Ecrire ("Entrez la valeur de B:")

 Lire (B)

 // Appel de la fonction Moyenne

 Ecrire ("La moyenne de A et B est : ", Moyenne(A, B))

Fin

Exercice 02 : Fonction $f(x) = 3x^3 + 4x + 8$

Écrire un algorithme qui contient la déclaration de fonction $f(x) = 3x^3 + 4x + 8$ et affiche le résultat de cette fonction pour $x = 1$, $x = 2$ et $x = 2,7$.

Solution :

Algorithme :

Algorithme Fonction_polynomiale

// Déclaration de la fonction $f(x) = 3x^3 + 4x + 8$

Fonction $f(x : \text{réel}) : \text{réel}$

Début

 Retourner $3*x^3 + 4*x + 8$

FinFonct

// Programme principal

Début

 //Appel de fonction

 Ecrire("f(1) = ", f(1), " f(2) = ", f(2), " f(2.7) = ", f(2.7))

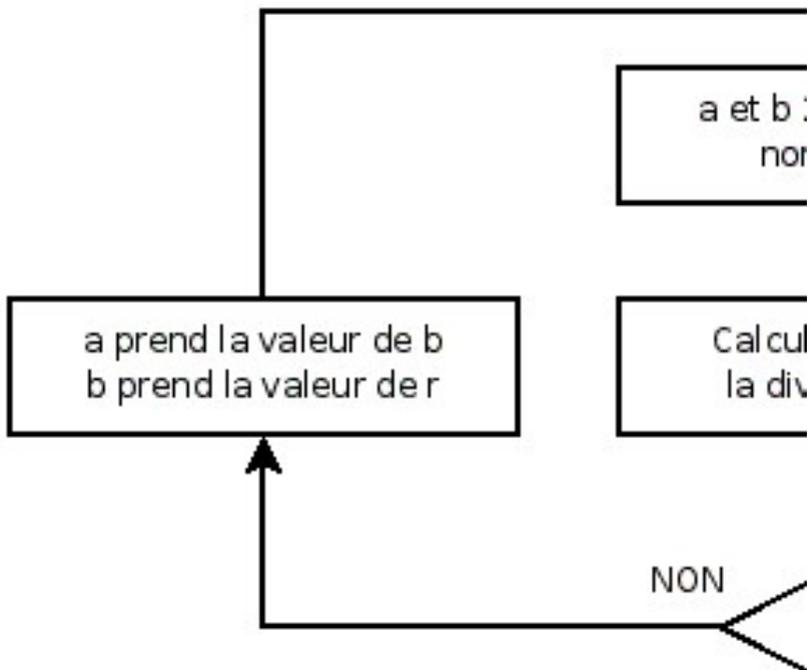
Fin

Exercice 03 : Fonction PGCD récursive

Ecrire un algorithme qui calcul le PGDC (plus grand diviseur commun) de deux nombre entier a et b non nuls ($a > b$) en utilisant une fonction récursive.

Solution :

La figure suivante présente le principe de calcul du PGDC de a et b.



Exemple :

$\text{PGCD}(5, 2) = 1$; $\text{PGCD}(10, 5) = 5$; $\text{PGCD}(12, 8) = 4$.

Algorithme :

Algorithme PGCD_récurive

Variable a, b : entier

// Déclaration de la fonction récursive qui calcul le PGCD

Fonction PGCD(x : entier, y : entier) : entier

Variable r : entier

Si y = 0 alors

Retourner x

Sinon

// r : reste de la division entière de x par y

r ← x mod y

Retourner PGCD(y, r)

FinSi

FinFonction

// Algorithme principal

Début

Ecrire((" Donnez a et b : "))

Lire(a, b)

Ecrire("PGCD(a, b) = ", PGCD(a, b))

Fin

Exercice 04 : Fonction somme récursive

Ecrire un algorithme qui calcul :

Somme(m) = 1 + 2 + 3 + 4 + ... + m-1 + m en utilisant une fonction récursive.

Solution :

Algorithme :

Algorithme Somme_réursive

Variable m : entier

// Déclaration de la fonction somme réursive

Fonction somme(n : entier) : entier

Début

Si n = 0 alors

Retourner 0

Sinon

Retourner somme(n - 1) + n

FinSi

FinFonct

// Programme principal

Début

Ecrire("Donnez m")

Lire(m)

Ecrire("La somme de m premiers entiers est : ", somme(m))

Fin

Exercice 05 : Passage des paramètres par adresse

Qu'affiche à l'écran l'algorithme suivant :

Algorithme Passage_par_adresse

Variables n, p : entiers

// Déclaration de la procédure échange

Procédure échange(**Var** a : entier, **Var** b : entier)

Variation c : entier

Début

Ecrire("Début échange ", a, " ", b)

c ← a

a ← b

b ← c

Ecrire("Fin échange ", a, " ", b)

FinProc

// Programme principal

Début

n ← 10

p ← 20

Ecrire("Avant l'appel ", n, " ", p)

échange n, p

Ecrire("Fin échange ", n, " ", p)

Fin

Solution :

Dans la ligne de code : *Procédure échange(Var a : entier, Var b : entier)*, il y a la présence du mot clé *Var*. Il s'agit donc d'un passage de paramètres par adresse.

Ce programme affiche à l'écran :

Avant l'appel 10 20

Début échange 10 20

Fin échange 20 10

Fin échange 20 10

Exercice 06 : Passage des paramètres par valeur

Soit l'algorithme suivant :

```
Algorithme passage_par_valeur
Variables n, p : entiers

// Déclaration de la procédure échange
Procédure échange(a : entier, b: entier)
Variable c : entier
Début
    Ecrire("Début échange ", a, " ", b)
    c ← a
    a ← b
    b ← c
    Ecrire("Fin échange ", a, " ", b)
FinProc

// Programme principal
Début
    n ← 10
    p ← 20
    Ecrire("Avant l'appel ", n, " ", p)
    échange n, p
    Ecrire("Fin échange ", n, " ", p)
Fin
```

- Quelle est la différence entre l'algorithme de l'exercice 05 et celui de l'exercice 06.
- Qu'affiche cet algorithme à l'écran

Solution :

La seule différence est l'absence du mot clé *Var* dans l'algorithme de l'exercice 06. Il s'agit d'un passage de paramètres par valeur dans ce dernier.

Ce programme affiche à l'écran :

```
Avant l'appel 10 20
Début échange 10 20
Fin échange 20 10
Fin échange 10 20
```

Remarque :

Les valeurs de n et p ne changent pas après l'appel de la procédure *échange* car le passage des paramètres est par valeur.

Exercice 07 : Suite numérique

Soit la suite numérique U_n suivante :

$$\begin{cases} \text{Si } n = 0, U_0 = 4 \\ \text{Si } n > 0, U_n = 5 U_{n-1} + 9 \end{cases}$$

Ecrire un algorithme qui calcul le terme U_n en utilisant une fonction récursive.

Solution :

Algorithme :

Algorithme suite_numérique

Variable n : entier

// Déclaration de la fonction U qui calcule le terme $U_m = U(m)$

Fonction $U(m$: entier) : réel

Si $m = 0$ alors

Retourner 4

Sinon

Retourner $5 * U(m - 1) + 9$

FinSI

FinFonction

// Algorithme principal

Début

Ecrire("Donner n ")

Lire(n)

Ecrire("Le terme U ", n , " = ", $U(n)$)

Fin

Exercice 08 : Partie entière

Ecrire un algorithme qui contient une fonction $E(x)$ qui renvoie la

partie entière d'un nombre réel saisi au clavier.

Exemple

$$E(2) = 2$$

$$E(-4) = -4$$

$$E(6.2) = 6$$

$$E(-10.2) = -11$$

Solution :

Soit x un nombre réel. Si $x \geq 0$ alors $E(x) = x \text{ div } 1$ (la partie entière de x est égal à la division entière de x par 1).

Si $x < 0$, il y a deux cas : si x est différent de $(x \text{ div } 1)$ dans ce cas $E(x) = (x \text{ div } 1) - 1$ sinon $E(x) = x \text{ div } 1$.

Algorithme :

Algorithme Partie_entière

Variable x : réel

// Déclaration de la fonction E : partie entière

Fonction E(y : réel) : entier

Variable r : entier

Si $y \geq 0$ alors

 //L'opérateur div : c'est la division entière

 Retourner $y \text{ div } 1$

Sinon

$r \leftarrow y \text{ div } 1$

 Si $y \neq r$ alors

 Retourner $r - 1$

 Sinon

 Retourner y

 FinSi

FinSi

FinFonct

// Programme principal

Début

 Ecrire("Donnez x : ")

 Lire(x)

 ‘ Appel de la fonction

 Ecrire("E(", x, ") = ", E(x))

Fin

Exercice 09 : Suite de Fibonacci

En utilisant une fonction récursive, écrire un algorithme qui détermine le terme U_n de la suite de Fibonacci définie comme suit :

$$\left| \begin{array}{l} U_0 = 0 ; U_1 = 1 \\ U_n = U_{n-1} + U_{n-2} , n \geq 2. \end{array} \right.$$

Solution :

L'algorithme :

```
Algorithme suite_fibonacci
Variable n : entier
// Déclaration de la fonction fibonacci
Fonction fibonacci(k : entier) : entier
    Si k = 0 alors
        Retourner 0
    SinonSI k = 1 alors
        Retourner 1
    Sinon
        Retourner fibonacci(k - 2) + fibonacci(k - 1)
    FinSi
FinFonction
// Programme principal
Début
    Ecrire("Donnez la valeur de n")
    Lire(n)
    Ecrire("Le terme U", n, " = ", fibonacci(n))
Fin
```

Exercice 10 : Fonction factorielle récursive

Ecrire un algorithme qui calcule la factorielle d'un nombre entier positif en utilisant la récursivité.

Solution :

Algorithme :

Algorithme calcul_factorielle

Variante n : entier

// Déclaration de la fonction Fact qui calcule la factorielle d'un nombre

Fonction Fact (m : entier) : entier

Début

Si m= 0 Alors

Retourner 0

Sinon

Retourner Fact(m-1) *m

FinSi

FinFonct

// Programme principal

Début

Ecrire((" Donner un entier n positif : "))

Lire(n)

// Appel de la fonction Fact

Ecrire("La factorielle de n est : ", fact(n))

Fin

Exercice 11 : Fonctions et structures

Soit :

- **Personne** une structure qui contient trois champs : nom, prénom et âge.

- **Etudiant1** et **Etudiant2** deux variables de type **Personne**.

- **Différence_age** est une fonction qui renvoie la différence d'âge entre l'étudiant1 et l'étudiant2.

Ecrire un algorithme qui permet de saisir l'âge de l'étudiant1 et l'étudiant2 et affiche la différence d'âge entre les deux étudiants (utiliser la fonction **Différence_age**).

Solution :

Algorithme :

Algorithme Différence_age

```
// Déclaration de la structure Personne
Type Structure Personne
    nom : chaîne
    prénom : chaîne
    age : entier
FinStruct
// Déclaration de deux variables de type Personne
Variables Etudiant1, Etudiant2 : Personne
// Déclaration de la fonction Différence_age
Fonction Différence_age (Etudiant1, Etudiant2 : Personne) : entier
Variable y : entier
Début
    Si Etudiant1.age > Etudiant2.age Alors
        y ← Etudiant1.age – Etudiant2.age
    Sinon
        y ← Etudiant2.age – Etudiant1.age
    FinSi
    Retourner y
FinFonction
// Début de l’algorithme
Début
    Ecrire("Entrez le nom, le prénom puis l'âge de l'étudiant 1 : ")
    Lire(etudiant1.nom, etudiant1.prénom, etudiant1.age)
    Ecrire("Entrez le nom, le prénom puis l'âge de l'étudiant 2 : ")
    Lire(etudiant2.nom, etudiant2.prénom, etudiant2.age)
    // Appel de la fonction Différence_age
    Ecrire("La différence d'âge entre ", etudiant1.nom, " et ",
    etudiant2.nom, " est :", Différence_age(Etudiant1, Etudiant2))
Fin
```