

Secteur Tertiaire Informatique
Filière « Etude et développement »

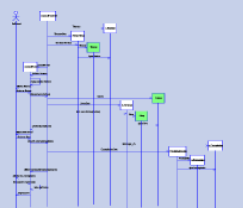
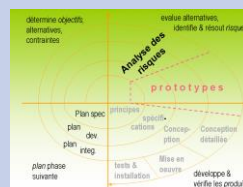
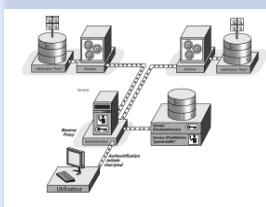
Ecrire un algorithme – Partie 1

Solutions Exercices Pseudocode

Apprentissage

Mise en situation

Evaluation



Préambule

Ce document propose la correction des exercices d'algorithmie du document « *Ecrire un algorithme - Partie 1 Exercices Pseudocode* ».

Les solutions exprimées sont écrites en **pseudocode uniquement**, en **conformité syntaxique et grammaticale** avec le support de cours « *Introduction à l'algorithmique* » de la séquence « *Ecrire un algorithme-Partie 1* »,

Il doit être complété par la réalisation des **mêmes exercices** du document « *Ecrire un algorithme – Partie 1 Exercices Langage de programmation* » pour lesquels les solutions sont fournies sous forme de projet **Java**.

Objectifs

A l'issue de la réalisation de ces exercices, vous saurez mettre en œuvre les solutions de problèmes informatiques exprimés en algorithmie et **pseudocode** et aurez assimilé tous les **fondamentaux** de la **programmation procédurale**.

Méthodologie

- Vous allez mettre en œuvre l'ensemble des notions et concepts liés à l'algorithmie dans la réalisation de ces exercices.
- Les notions d'algorithmie maîtrisées à l'issue de ces exercices en pseudocode, vous écrirez ensuite les solutions exprimées dans le **langage de programmation précisé par votre formateur**.
- Vous les ferez alors valider par votre formateur.
- Vous constaterez que, une fois la solution des exercices exprimée en **pseudocode**, la conversion vers **un langage de programmation** n'est qu'une formalité.
- La seule contrainte étant le **respect de la syntaxe** imposé par le langage.

Exercice N°1

Solution avec TantQue

Programme PremiersNombresEntiers

// Ce programme calcule la somme des N premiers nombres entiers avec la structure itérative **TANTQUE**

Variables

N : **entier**

resultat : **entier**

cpt : **entier**

Début

resultat := 0

cpt := 1

N := lireEntier()

tantque (cpt <= N) **faire**

 resultat := resultat + cpt

 cpt := cpt + 1

fintantque

écrire (« Le résultat avec tantque est : », resultat)

Fin

Solution avec Répéter

Programme PremiersNombresEntiers

// Ce programme calcule la somme des N premiers nombres entiers avec la structure itérative **REPETER**

Variables

N : **entier**

resultat : **entier**

cpt : **entier**

Début

resultat := 0

cpt := 1

N := lireEntier()

répéter

resultat := resultat + cpt

cpt := cpt + 1

jusqu'à (cpt > N)

écrire (« Le résultat avec répéter est : », resultat)

Fin

Solution avec Pour

Programme PremiersNombresEntiers

// Ce programme calcule la somme des N premiers nombres entiers avec la structure itérative **POUR**

Variables

N : **entier**

resultat : **entier**

cpt : **entier**

Début

N := lireEntier()

pour (resultat := 0, cpt := 1; cpt <= N ; cpt := cpt + 1) **faire**

 resultat := resultat + cpt

finpour

écrire (« Le résultat avec pour est : », resultat)

Fin

Exercice N°2

Solution avec TantQue

Programme Factorielle

// Ce programme calcule la factorielle de l'entier N avec la structure itérative **TANTQUE**

Variables

N : **entier**

resultat : **entier**

cpt : **entier**

Début

resultat := 1

N := lireEntier()

cpt := N

Si (N = 0 **OU** N = 1) **Alors**

écrire (« La factorielle de », N, « avec tantque est : », resultat)

Sinon

tantque (cpt > 1) **faire**

 resultat := resultat * cpt

 cpt := cpt - 1

fintantque

écrire (« La factorielle de », N, « avec tantque est : », resultat)

Finsi

Fin

Solution avec Répéter

Programme Factorielle

// Ce programme calcule la factorielle de l'entier N avec la structure itérative **REPETER**

Variables

N : **entier**

resultat : **entier**

cpt : **entier**

Début

resultat := 1

N := lireEntier()

cpt := N

Si (N = 0 **ou** N = 1) **Alors**

écrire (« Le résultat avec répéter est : », resultat)

Sinon

répéter

 resultat := resultat * cpt

 cpt := cpt - 1

jusqu'à (cpt = 1)

écrire (« Le résultat avec répéter est : », resultat)

Finsi

Fin

Solution avec Pour

Programme Factorielle

// Ce programme calcule la factorielle de l'entier X avec la structure itérative POUR

Variables

N : **entier**

resultat : **entier**

cpt : **entier**

Début

resultat := 1

N := lireEntier()

Si (N = 0 **ou** N = 1) **Alors**

écrire (« Le résultat avec pour est : », resultat)

Sinon

pour (cpt := N ; cpt > 1 ; cpt := cpt - 1) **faire**

 resultat := resultat * cpt

finpour

écrire (« Le résultat avec pour est : », resultat)

Finsi

Fin

Exercice N°3

Programme EquationSecondDegre

// Ce programme calcule et affiche les solutions d'une équation du second degré.

Variables

a : **entier**
b : **entier**
c : **entier**
delta : **réel**
resultat : **réel**

Début

écrire (« Saisir la valeur de a : »)

a := lireEntier()

écrire (« Saisir la valeur de b : »)

b := lireEntier()

écrire (« Saisir la valeur de c : »)

c := lireEntier()

delta := (b*b) - (4*a*c)

Si (delta < 0) **Alors**

écrire (« Il n'y a pas de solution pour cette équation »)

Sinon

Si (delta = 0) **Alors**

resultat := -b/(2*a)

écrire (« Il y a une solution double pour cette équation : »)

écrire (resultat)

Sinon

écrire (« Il y a 2 solutions pour cette équation : »)

écrire (-b + ($\sqrt{\text{delta}/2*a}$))

écrire (-b - ($\sqrt{\text{delta}/2*a}$))

Finsi

Finsi

Fin

Exercice N°4

Programme CalculPuissance

// Ce programme calcule et affiche **X puissance Y**.

Variables

x : **entier**

y : **entier**

resultat : **entier**

Début

écrire (« Saisir la valeur de x : »)

x := lireEntier()

écrire (« Saisir la valeur de y : »)

y := lireEntier()

Si (y=0) **Alors**

écrire (« Le résultat est 1 »)

Sinon

Si (x=0) **Alors**

écrire (« Le résultat est 0 »)

Sinon

 resultat := puissance (x,y)

écrire (« Le résultat est », resultat)

Finsi

Finsi

Fin

fonction puissance (**entrée** : x : entier, **entrée** y : entier) : entier

// Cette fonction calcule et retourne x élevé à la puissance y.

Variables

compteur : **entier**

resultat : **entier**

Début

compteur := 1

resultat := 1

tantque (compteur <= y) **faire**

 resultat := resultat * x

 compteur := compteur + 1

fintantque

retourner (resultat)

Fin

Exercice N°5

Programme RechercheDichotomique

// Ce programme effectue une **recherche dichotomique**, dans un tableau d'entiers **déjà trié**.
// Si l'entier recherché est trouvé, on affiche le rang où il se trouve dans le tableau, sinon on indique que l'entier n'existe pas dans le tableau.

types tabent = tableau[10] de **entier** // Création du type *tabent*

Variables

x : **entier**
position: **entier**
resultat : **entier**
tabEntiers : tabent

Début

tabEntiers = { -2, -1, 0, 13, 24, 37, 44, 56, 99, 117}; // Tableau trié

écrire (« Veuillez saisir la valeur de X à rechercher dans le tableau »)

x := lireEntier()

position =rechercherEntier (tabEntiers, x);

Si (position == -1) **Alors**

écrire(« x n'existe pas dans le tableau »)

Sinon

écrire(« x se trouve à la position », position, « dans le tableau »)

Fin

Fin

fonction rechercherEntier (**entrée** tab : tabent, **entrée** x : entier) : entier
// Cette fonction recherche la valeur de **x** dans le tableau **tab** et retourne sa position.
// Si y n'existe pas, la fonction retourne -1

Variables

indiceBas : **entier**
indiceHaut : **entier**
indiceMilieu : **entier**
trouvé : **booléen**

Début

indiceBas := 0
indiceHaut := tab.**taille** - 1
indiceMilieu := (indiceBas + indiceHaut)/2

trouvé := **faux**

tantque (trouvé = **faux** **ET** indiceBas <= indiceHaut) **faire**

Si (y < tab[indiceMilieu]) **Alors**

 indiceHaut := indiceMilieu - 1

Sinon Si (y > tab[indiceMilieu]) **Alors**

 indiceBas := indiceMilieu + 1

Sinon faire

 trouvé := **true**

Finsi

Finsi

 indiceMilieu := (indiceBas + indiceHaut) / 2

fintantque

Si (trouvé = **vrai**) **Alors**

retourner (indiceMilieu)

Sinon

retourner (-1)

Finsi

Exercice N°6

Programme TableauFactorielles

// Ce programme range dans un tableau d'entiers à deux dimensions les entiers de 1 à 10 et leur factorielle respective.

types tabent = **tableau**[10] **de entier**
matrice = **tableau**[2] **de** tabent
indiceColonne = **entier**
x = **entier**

Variables

TableauFactorielles : matrice

Début

// Remplissage de la **première ligne** du tableau

indiceColonne := 0

tantque (indiceColonne < TableauFactorielle[0].**taille**) **faire**

TableauFactorielle[0] [indiceColonne] = indiceColonne + 1

fintantque

// Remplissage de la **deuxième ligne** du tableau

indiceColonne := 0

tantque (indiceColonne < TableauFactorielle[0].**taille**) **faire**

TableauFactorielle[1] [indiceColonne] = CalculFactorielle(indiceColonne + 1)

fintantque

écrire (« Pour quelle valeur entière souhaitez-vous calculer la factorielle ? (entre 1 et 10) »)

x := lireEntier()

Si (x < 1 **OU** x > 10) **Alors**

écrire(« x doit être compris entre 1 et 10 »)

Sinon

resultat := TableauFactorielle[1][x-1]

écrire(« La factorielle de » , x , « est : », resultat)

Fin

Fin

CREDITS

ŒUVRE COLLECTIVE DE l'AFPA

Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services

Equipe de conception (IF, formateur, mediatiseur)

Chantal Perrachon – IF Neuilly-sur-Marne

Michel Coulard – Formateur Evry

Date de mise à jour : 05/01/15

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »