# Algorithmie

## Introduction

Un Algorithme est une suite d’instructions à résoudre un problème.

Exemple :

Problème : j’ai envie de faire pipi

Solution : Aller aux toilettes

## Pseudo code / Algobox

Début algo

Se lever

Aller aux toilettes

Se déshabiller

PISSER

Tirer la chasse

Se rhabiller

Se laver les mains

Fin algo

Variables

Ingredient1

Ingredient2

Ingredient3

Début Algo

Sortir ingredient1

Sortir ingredient2

Mélanger ingredient1 avec ingredient2 donne ingredient3

Mettre ingredient3 au four

Sortir la préparation

Fin algo

Variables

chiffre1 : entier

chiffre2 : entier

resultat : entier

Debut Algo

chiffre1 = Saisir « Le premier chiffre ? »

chiffre2 = Saisir « Le deuxième chiffre ? »

resultat = chiffre1 \* chiffre2

Afficher resultat

Fin algo

Cf 1-multiplication.alg

## Variables

Les variables dans un programme sont typées. Types primitifs :

* Chaines de caractères
* Entier
* Décimal
* Booléan

Dans les langages comme PHP ou javascript (faiblement typé), ce n’est pas obligatoire de typer les variables.

Dans d’autres langages comme Java (fortement typé), c’est obligatoire.

## Structures conditionnelles

## Le SI/SINON (le IF/ELSE in english)

Tester une condition, puis exécuter des instructions si la condition est vraie, avec la possibilité d’exécuter d’autres instructions est fausse.

Variables

age : entier

Debut Algo

age = Saisir « Quel est votre âge ? » : LIRE age

SI age >= 18

Afficher « Vous êtes majeur »

SINON

Afficher « Vous êtes mineur »

FIN SI

Fin algo

Cf 2-isMajeur.alg

On peut cumuler les conditions grâce aux opérateurs logiques ET et OU.

On peut prioriser les OU grâce aux parenthèses.

Exemple :

Debut Algo

SI 5 == 5 ET 6 < 10

// on rentre

Afficher « 5 est égal à 5 ET 6 est inférieur à 10

FIN SI

SI 5 == 5 OU 6 > 10

// on rentre

Afficher « 5 est égal à 5 OU 6 est supérieur à 10

FIN

SI 5 == 5 ET 3 > 5 OU 9 == 9

// on rentre : le premier ensemble lié par un ET est faux, mais ensuite le deuxième est vrai. Comme on a un OU, l’ensemble des conditions est vrai.

FIN SI

SI 5 == 5 ET (3 > 5 OU 9 == 8)

// on rente pa

FIN SI

Fin algo

## Le SELON (switch in english)

Variable :

note : entier

Debut Algo

note = saisir « Quelle note as-tu eue ? »

SELON note

CAS 5 :

Afficher « c’est pas très bien »

CAS 10 :

CAS 11 :

CAS 12 :

Afficher « c’est moyen »

CAS 20 :

Afficher « pas mal »

DEFAUT :

Afficher « laisse moi tranquille avec tes notes de merde »

FIN SELON

Fin algo

## Le SINON SI

Au lieu d’écrire ce code pas lisible (et degueu, soyons clair)

Debut algo

SI note == 5

Afficher « »

SINON

SI note == 10 ou note == 11 ou note == 12

Afficher « »

SINON

SI note == 20

Afficher « »

SINON

Afficher « »

FIN SI

FIN SI

FIN SI

Fin algo

Debut algo

SI note == 5

Afficher « »

SINON SI note == 10 ou note == 11 ou note == 12

Afficher « »

SINON SI note == 20

Afficher « »

SINON

Afficher « »

FIN SI

Fin algo

## La concaténation

Variables :

Chaine1, chaine2, chaine3 : chaines de caractères

Debut Algo

Chaine1 = «Bonjour»

Chaine2 = «tout»

Chaine3 = «le monde»

Afficher chaine1 CONCAT chaine2 CONCAT chaine3

// affiche «bonjourtoutle monde»

Afficher chaine1 CONCAT « » CONCAT chaine2 CONCAT « » CONCAT chaine3 CONCAT « !»

// affiche «bonjour tout le monde !»

Fin algo

## Exercice

Ecrire l’algo qui demande à l’utilisateur un chiffre entre 1 et 9, sinon message erreur et fin programme.

S’il a bien saisi un chiffre correct, afficher la table de multiplication sous ce format :

1 \* 2 = 2

2 \* 2 = 4

…

9 \* 2 = 18

Variables :

Chiffre, resultat, resultat2, resultat3 : entier

Debut algo

chiffre = saisir « Veuillez saisir un chiffre entre 1 et 9 »

SI chiffre < 1 OU chiffre > 9

Afficher « Mauvaise saisie »

SINON

// affichage de la table

resultat = 1 \* chiffre

afficher « 1 \* » CONCAT chiffre CONCAT « = » CONCAT resultat

resultat2 = 2 \* chiffre

afficher « 2 \* » CONCAT chiffre CONCAT « = » CONCAT resultat2

resultat3 = 3 \* chiffre

afficher « 3 \* » CONCAT chiffre CONCAT « = » CONCAT resultat3

// la même pour les 6 lignes restantes de la table de multiplication

FIN SI

Fin algo

Cf 3-table-multiplication.alg

## Structures itératives

POUR (FOR) / TANT QUE (WHILE) / REPETER JUSQU’A (DO WHILE)

* Boucle POUR

Debut algo

// cette boucle affiche 950 fois le message

POUR i de 0 à 949 PAR PAS DE 1

Afficher « Je dois pas tricher »

FIN POUR

// cette boucle afficher 475 fois le message

POUR i de 0 à 949 PAR PAS DE 2

Afficher « Je dois pas tricher »

FIN POUR

Fin algo

* TANT QUE

Debut Algo

// même exemple que pour la boucle POUR (attention le contexte fait que la boucle pour est à privilégier ici)

i = 0

TANT QUE i <= 949

Afficher « Je dois pas tricher »

i = i + 1 // incrémenter i : la nouvelle valeur est égale à la valeur actuelle + 1

FIN TANT QUE

// exemple où la boucle TANT QUE est préconisée

chiffre = saisir « Veuillez saisir un chiffre entre 1 et 9 »

TANT QUE chiffre < 1 OU chiffre > 9

chiffre = saisir « Veuillez saisir un chiffre entre 1 et 9 »

FIN TANT

// autre exemple avec nombre de tours inconnus

somme = 0

chiffreAleatoire = RANDOM(1, 10)

somme = somme + chiffreAleatoire

i = 0

TANT QUE somme < 100

chiffreAleatoire = RANDOM(1, 10)

somme = somme + chiffreAleatoire

i = i + 1

FIN TANT QUE

Afficher « Nombre de tours de boucle : » CONCAT i

Fin algo

* REPETER JUSQU’A

Debut algo

i = 0

REPETER

Afficher « Je dois pas tricher »

i = i + 1

TANT QUE i <= 949

REPETER

chiffre = saisir « Veuillez saisir un chiffre entre 1 et 9 »

TANT QUE chiffre < 1 OU chiffre > 9

Fin algo

Boucle POUR : à utiliser quand on connait précisément le nombre d’itération à faire

Boucle TANT QUE : quand on ne sait pas le nombre d’itération à l’avance et qu’il est possible qu’il n’y en ait aucune

Boucle REPETER : quand on ne sait pas à l’avance le nombre d’itération mais qu’il y en a au moins une

* Table de multiplication avec boucle

Variables :

chiffre, resultat: entier

Debut algo

REPETER

chiffre = saisir « Veuillez saisir un chiffre entre 1 et 9 »

TANT QUE chiffre < 1 OU chiffre > 9

// arrivé à cette ligne de code, la variable chiffre est forcément comprise entre 1 et 9

// affichage de la table : 9 itérations (une pour chaque ligne de la table de multiplication)

POUR i de 1 à 9 PAR PAS de 1

resultat = i \* chiffre

afficher i CONCAT « \* » CONCAT chiffre CONCAT « = » CONCAT resultat

FIN POUR

Fin algo

* Exercices

1 - Demander à l’utilisateur de saisir 5 notes

Afficher la moyenne de ces notes

* + Version sans boucle
  + Version avec boucle

Variables

note1 : entier

note2 : entier

note3 : entier

note4 : entier

note5 : entier

note : entier

somme : entier

moyenne : nombre

Debut algo

// 1- version sans boucle

Lire note1

Lire note2

Lire note3

Lire note4

Lire note5

somme = note1 + note2 + note3 + note4 + note5

moyenne = somme / 5

afficher moyenne

// sans variable intermédiaire

moyenne = (note1 + note2 + note3 + note4 + note5) / 5

afficher moyenne

// afficher directement la moyenne

Afficher (note1 + note2 + note3 + note4 + note5) / 5

// 1bis – Code qui permet de visualiser qu’une boucle est faisable

somme = 0

Lire note

somme = somme + note

Lire note

somme = somme + note

Lire note

somme = somme + note

Lire note

somme = somme + note

Lire note

somme = somme + note

moyenne = somme / 5

// 2 – version avec boucle

somme = 0

POUR i de 0 à 4 PAR PAS de 1

Lire note

somme = somme + note

FIN POUR

moyenne = somme / 5

Fin algo

2- Demander à l’utilisateur de saisir des notes.

C’est lui qui décide quand la saisie s’arrête en tapant -1

Afficher la moyenne de ces notes

Variable

Debut algo

// VERSION REPETER

somme = 0

nbNotes = 0

REPETER

Lire note

SI note != -1

nbNotes = nbNotes + 1

somme = somme + note

FIN SI

TANT QUE note != -1

moyenne = somme / nbNotes

// VERSION TANT QUE

somme = 0

nbNotes = 0

Lire note

TANT QUE note > -1

nbNotes = nbNotes + 1

somme = somme + note

Lire note

FIN TANT QUE

moyenne = somme / nbNotes

Fin algo

## Exercices

### Exercice 1

Demander à l’utilisateur deux nombres, et afficher le résultat de la soustraction.

### Exercice 2

Demander à l’utilisateur deux nombres compris entre 5 et 50, et afficher le résultat de la multiplication.

### Exercice 3

Demander à l’utilisateur deux nombres et afficher le résultat de la division. Vérifiez d’abord si la division est possible.

### Exercice 4

Demander à l’utilisateur son âge, et afficher s’il est majeur ou mineur. S’il est mineur, afficher combien d’années il reste avant qu’il soit majeur.

### Exercice 5

Demander à l’utilisateur un nom de fruits, lui afficher

* « Cool » dans le cas où il dit « fraise »
* « Pas cool » dans le cas où il dit « banane »
* « Etrange » dans le cas où il dit « kiwi »
* « Comme tu veux » dans tous les autres cas

### Exercice 6

### Exercice 7

### Exercice 8

### Exercice 9