

Secteur Tertiaire Informatique
Filière « Etude et développement »

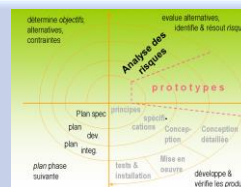
Ecrire un algorithme – Partie 1

Exercices Pseudocode

Apprentissage

Mise en situation

Evaluation



Préambule

Afin de réaliser ces **exercices** de mise en œuvre, vous devez parfaitement connaître les bases de l'algorithmie :

- La notion de **variable**.
- Les **structures de contrôle**.
- Les **structures itératives**.
- Les **tableaux**.
- Les **procédures et fonctions**.
- Les **paramètres et retour** de fonctions.

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

Dans un premier temps, à travers ces exercices, vous écrirez d'abord les solutions exprimées dans le **langage algorithmique** ou **pseudocode** puis les ferez valider par votre formateur.

Pour chaque solution, fournissez un jeu d'essais et vérifiez le bon comportement de vos algorithmes.

Dans un deuxième temps, avec des **énoncés quasis-identiques** mais orientés **programmation concrète** (document ***M-Exercices-programmation-procedurale-serie1***), vous coderez ces solutions dans le langage de votre choix en utilisant l'EDI et le langage indiqués par votre formateur.

Exercice N°1

- Ecrivez un algorithme en **pseudocode** permettant le calcul de la somme des N premiers nombres entiers.
- La variable N sera saisie par l'utilisateur grâce à une fonction *lireEntier()* qui permet de :
 - saisir un entier au clavier et ...
 - renvoyer cette valeur en retour.
- Elle s'utilise donc de la manière suivante :

Variables

N : **entier** // Déclaration de la variable N de type entier

$N := \text{lireEntier}()$ // Lecture d'un entier et affectation de la variable N

.....

- Par exemple : A l'issue de la saisie de la valeur 6 et après validation, la variable N vaut 6.
- Vous mettrez en œuvre cet algorithme en générant **3 solutions** en **pseudocode** : une pour chaque structure itérative, **TANT QUE**, **REPETER** et **POUR**.
- L'affichage du résultat se fera ainsi pour chaque solution :

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

Exercice N°2

- Ecrivez un algorithme en **pseudocode** permettant le calcul de la factorielle d'un **entier** X .
- La variable X sera saisie selon le même procédé que ci-dessus.
- Vous écrirez cet algorithme avec les 3 structures itératives **TANT QUE**, **REPETER** et **POUR** ...
- Les affichages seront réalisés avec la méthode que ci-dessus.

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

Exercice N°3

- Ecrivez un algorithme en **pseudocode** permettant la résolution d'une équation du second degré $ax^2 + bx + c = 0$.
- On saisira les **entiers** a , b et c . avec la fonction *lireEntier()*.
- La(les) solution(s) seront de type **réel**.
- Fournissez une seule solution avec les **structures itératives** que vous jugerez pertinentes.

Exercice N°4

- Ecrivez un algorithme en **pseudocode** permettant le calcul d'un nombre X élevé à la puissance Y .
- Les **entiers positifs** X et Y seront saisis toujours avec la fonction *lireEntier()*.
- Ecrivez une fonction *entier puissance(entier x , entier y)* qui admet deux paramètres x et y de type **entier** et qui renvoie le résultat de x élevé à la puissance y .
- La fonction *puissance* sera appelée au sein du programme principal. La valeur retournée sera ensuite affichée pour produire le résultat.

Exercice N°5

- Ecrivez un algorithme en **pseudocode** permettant la recherche **dichotomique** d'une variable entière X dans un tableau d'entiers *TabEntiers* de taille TAILLE_MAX **déjà trié**.
- Les principes de la recherche **dichotomique** consistent à diviser par deux l'espace de recherche par deux, tant que l'on n'a pas trouvé X , et à y rechercher la valeur souhaitée, après avoir changé les bornes de l'espace de recherche.
- Naturellement, ce principe ne fonctionne que si le tableau est déjà trié.
- Demandez des informations complémentaires à votre formateur si besoin.
- Utilisez les structures **itératives** et de contrôle que vous jugerez nécessaires.
- Déclarez un tableau de 10 entiers. Rangez-y 10 valeurs négatives, positives ou nulles dans l'**ordre croissant**.
- Ecrivez une fonction *entier rechercherEntier (entier $tab[]$, entier x)* qui recherche la **position** de la variable x dans le tableau tab en mettant en œuvre la recherche dichotomique.
- Si x est trouvée dans tab , la fonction renvoie le rang du tableau auquel elle se trouve.

- Si x n'existe pas dans le tableau, la fonction renvoie -1 au programme appelant, indiquant ainsi l'absence de cette valeur.

-2	-1	0	13	24	37	44	56	99	117
----	----	---	----	----	----	----	----	----	-----

Dans le tableau ci-contre, en recherchant la valeur 44, la fonction *rechercheEntier(...)* retourne la valeur 7 : c'est le rang dans le tableau où elle a été trouvée.

Le résultat produira l'affichage suivant :

44 se trouve au rang 7 dans le tableau
--

NB : On peut connaître la **taille** et donc le **nombre d'éléments d'un tableau** en utilisant la **notation pointée** suivi du mot-clé **taille**.

Par exemple la valeur de *tab.taille*, appliquée au tableau ci-dessous est **10**.

Exercice N°6

- Ecrire un algorithme en **pseudocode** permettant le rangement dans un tableau d'entiers à deux dimensions «*TableauFactorielles*» des 10 premiers entiers numériques et de leur factorielle respective.
- La première ligne sera valorisée avec les entiers de 1 à 10.
- La seconde sera valorisée avec la factorielle de chaque entier correspondant à la ligne au-dessus.

1	2	3	4	5	6	7	8	9	10
1	2	6	24	120	720	5040	40320	362880	3628800

- Pour calculer (avant de ranger) la factorielle de chaque entier de 1 à 10, on fera appel à une fonction baptisée «*CalculFactorielle*» permettant de renvoyer au module appelant principal la factorielle de l'entier transmis en tant qu'argument ou paramètre.
- L'appel se fera donc de la manière suivante :

TableauFactorielles[1][i] = CalculFactorielle(i+1) ;

- Le remplissage des deux lignes du tableau se fera dans une même et unique boucle.
- On exploitera ensuite le tableau pour répondre à la question suivante :

Pour quelle valeur souhaitez-vous calculer la factorielle (de 1 à 10 max) ?

... qui produira l'affichage suivant :

La factorielle de 5 est : 120

Version	Date	Auteur(s)	Action(s)
1.0	05/11/14	M. Coulard	Création du document
1.1	05/01/15	C. Perrachon	Révision du document
1.2	30/01/15	M.Coulard	Reformulation de l'énoncé pour que seul le pseudocode figure dans la production des exercices. Le fichier a été renommé : <i>Exercices algorithmie-pseudocode.docx</i>

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 : 30/01/15

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. »