



08/11/2022

ALGORITHMIQUE

Exercices : généralités



Rudy Lesur
SOFIP

Table des matières

<u><i>I Somme des n premiers nombre entiers</i></u>	<u>2</u>
<u><i>II Factorielle des n premiers nombre entiers</i></u>	<u>3</u>
<u><i>III Résolution d'une équation du second degré</i></u>	<u>4</u>
<u><i>IV Calcul de X à la puissance Y</i></u>	<u>5</u>
<u><i>V Recherche dichotomique d'une valeur dans un tableau</i></u>	<u>6</u>

I Somme des n premiers nombre entiers

Ecrivez un algorithme en pseudocode permettant le calcul de la somme des N premiers nombres entiers.

Si par exemple on saisit 3 l'algorithme devra réaliser le calcul de $1+2+3$ qui nous donne 6.

Etapes à réaliser :

- La variable N sera saisie par l'utilisateur grâce à une fonction **lireEntier()**, **il ne vous est pas demandé d'écrire la fonction lire entier mais de juste l'utiliser.**
Cette fonction permet de saisir un entier au clavier et renvoyer cette valeur en retour.
Elle s'utilise donc de la manière suivante dans le programme principal :

```
variable N : entier           // Déclaration de la variable N de type entier  
N := lireEntier()             // Lecture d'un entier et affectation de la variable N
```

- Ecrire ensuite en utilisant une boucle la solution pour le calcul de la somme. Créer autant de variables que nécessaires.
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 exemple avec la boucle Tantque :
écrire (« Le résultat avec tantque est : », resultat)

Jeux de test pour votre algorithme :

- N vaut 0 le résultat attendu est 0.
- N vaut 1 le résultat attendu est 1.
- N vaut 2 le résultat attendu est 3.
- N vaut 3 le résultat attendu est 6.
- N vaut 4 le résultat attendu est 10
- N vaut 5 le résultat attendu est 15.
- N vaut 6 le résultat attendu est 21.

Aide pour l'exercice :

- Dans cet exercice nous ne travaillons que les structures répétitives, pas de structures conditionnelles

II Factorielle des n premiers nombre entiers

Ecrivez un algorithme en pseudocode permettant le calcul de la factorielle des N premiers nombres entiers.

Si par exemple on saisit 4 l'algorithme devra réaliser le calcul de $1*2*3*4$ qui nous donne 24.

Etapes à réaliser :

- La variable N sera saisie par l'utilisateur grâce à une fonction **lireEntier()**, **il ne vous est pas demandé d'écrire la fonction lire entier mais de juste l'utiliser.**
Cette fonction permet de saisir un entier au clavier et renvoyer cette valeur en retour.
Elle s'utilise donc de la manière suivante dans le programme principal :

```
variable N : entier                // Déclaration de la variable N de type entier  
N := lireEntier()                  // Lecture d'un entier et affectation de la variable N
```

- Ecrire ensuite en utilisant une boucle la solution pour le calcul de la factorielle. Créer autant de variables que nécessaires.
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 exemple avec la boucle Tantque :
écrire (« La factorielle de », X, « avec tantque est : », resultat)

Jeux de test pour votre algorithme :

- N vaut 0 le résultat attendu est 1.
- N vaut 1 le résultat attendu est 1.
- N vaut 2 le résultat attendu est 2.
- N vaut 3 le résultat attendu est 6.
- N vaut 4 le résultat attendu est 24
- N vaut 5 le résultat attendu est 120.
- N vaut 6 le résultat attendu est 720.

Aide pour l'exercice :

- Dans cet exercice nous travaillons les structures répétitives, et les structures conditionnelles pour les valeurs de N correspondant à 0 ou 1 qui par convention donne le même résultat : 1.

III Résolution d'une équation du second degré

Ecrivez un algorithme en pseudocode permettant la résolution d'une équation du second degré qui est de type $ax^2 + bx + c = 0$.

Si par exemple l'équation est $4x^2 + 4x + 1 = 0$ alors le résultat de l'algorithme est une solution réelle double $x = -1/2$

Etapes à réaliser :

- Les variables a, b et c seront saisies par l'utilisateur grâce à une fonction **lireEntier()**.
- Déclarer la ou les solutions sont de type réel.
- Vérifier que la valeur de « a » saisie soit supérieur à 0 sinon il y a une erreur (A gérer, soit par un message d'erreur ou forcer l'utilisateur à saisir une valeur pour « a » non égale à 0).
- Déclarer une variable « D » qui sera le discriminant. Ce discriminant sera calculé et égal au résultat de l'opération : $b^2 - 4ac$ (pensez à écrire l'opération simplement, une valeur au carré est le résultat de la multiplication d'une valeur par elle-même).
- Il faut ensuite contrôler la valeur de « D ».

Si $D = 0$ alors la solution sera unique $x = -\frac{b}{2a}$

Si $D > 0$ alors il y aura 2 solutions $x_1 = \frac{-b - \sqrt{D}}{2a}$ et $x_2 = \frac{-b + \sqrt{D}}{2a}$

Si $D < 0$ alors il n'y a pas de solution réelle.

- L'affichage du résultat se fera ainsi pour chaque solution
(« Il y a une solution double pour cette équation $x =$: », x) ou
(« Il y a 2 solutions pour cette équation : $x_1 =$ » x_1 « et $x_2 =$ » x_2) ou
(« Il n'y a pas de solution pour cette équation »)

Jeux de test pour votre algorithme :

- $a=4, b=4, c=1$ le résultat attendu est : Il y a une solution double : $x = -1/2$
- $a=3, b=5, c=7$ le résultat attendu est : Il n'y a pas de solution pour cette équation
- $a=2, b=9, c=-5$ le résultat attendu est : Il y a 2 solutions : x_1 et x_2 ...
- $a=0$ le résultat attendu sera de votre choix, soit une phrase indiquant que « a » ne doit pas être nul sinon nous n'avons pas une équation du second degré ou alors si vous forcez l'utilisateur à saisir un « a » non nul par le biais d'une boucle.

Aide pour l'exercice :

- Dans cet exercice nous travaillons les structures répétitives, et les structures conditionnelles. Pour le contrôle de la valeur de D vous pouvez utiliser des **Si Alors** ou alors utiliser une imbrication de **Si Alors Sinon Si Alors**

IV Calcul de X à la puissance Y

Ecrivez un algorithme en pseudocode permettant le calcul d'un nombre X élevé à la puissance Y

Si par exemple on saisit 3 pour X et 2 pour Y on obtient comme résultat 9.

Etapes à réaliser :

- Les variables X et Y seront saisies par l'utilisateur grâce à une fonction **lireEntier()**. Si X vaut 0 alors le résultat sera directement 0 (pas besoin de calculer) Si X vaut 1 alors le résultat sera directement 1 (idem pas besoin de calculer)
- Déclarer le résultat de type entier
- Ecrire une fonction puissance qui prendra comme paramètres d'entrée X et Y et qui renvoie une valeur de sortie de type entier. Cette fonction fera le calcul 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

Jeux de test pour votre algorithme :

- X = 4 et Y = 3 le résultat attendu est : 64
- X = 1 et Y = 5 le résultat attendu est : 1
- X = 0 et Y = 2 le résultat attendu est : 0

Aide pour l'exercice :

- Dans cet exercice nous travaillons les structures répétitives, et les structures conditionnelles. Nous utiliserons aussi le principe des fonctions. Pensez donc à écrire la définition de la fonction avant de l'utiliser dans le programme principal. Pensez également à vérifier que X et Y soient des entiers positifs.

V Recherche dichotomique d'une valeur dans un tableau

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, 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.

C'est le principe du Juste prix avec comme informations prix plus grand ou plus petit que la valeur donnée.

Etapes à réaliser :

- Déclarez un tableau de 10 entiers. Rangez-y 10 valeurs négatives, positives ou nulles dans l'ordre croissant.
- Déclarer le résultat de type entier
- Ecrivez une fonction rechercherEntier qui prendra comme paramètre d'entrée le tableau d'entier TabEntiers et l'entier X à chercher. Cette fonction aura comme valeur de retour un entier dont sa valeur dépend du résultat de la recherche
- 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.
- La fonction puissance sera appelée au sein du programme principal. La valeur retournée sera ensuite affichée pour produire le résultat

Jeux de test pour votre algorithme :

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

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

Aide pour l'exercice :

- 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-dessus est 10.
- Il faut dans un premier temps identifier le milieu du tableau et comparer la valeur à X. Si la valeur est égale alors on a le rang qui est celui du milieu sinon si la valeur est plus grande alors on parcourt que la deuxième partie du tableau à partir du milieu jusqu'à la fin et ainsi de suite. Si la valeur est plus petite on parcourt la première partie du tableau à partir du début du tableau jusqu'à la valeur du milieu. Il faut bien sûr redéfinir les nouvelles valeurs de début fin et milieu après chaque comparaison