

Exercices Algorithmique

Benjamin Dallard

April 2019

1 Bases

1.1 Simple échange

Ecrire un algorithme permettant d'échanger les valeurs de deux variables A et B.

1.2 Le carré

Ecrire un programme qui demande un nombre à l'utilisateur, puis qui calcule et affiche le carré de ce nombre.

1.3 La condition

Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on laisse de côté le cas où le nombre vaut zéro).

1.4 Le produit

Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif (on laisse de côté le cas où le produit est nul). Attention toute fois : on ne doit pas calculer le produit des deux nombres.

1.5 Calcul

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

1.6 Additivité

Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer : $1 + 2 + 3 + 4 + 5 = 15$.

2 Les tableaux

2.1 La somme

Ecrire un algorithme calculant la somme des valeurs d'un tableau (on suppose bien sur que le tableau a été préalablement saisi).

2.2 La somme : retour

Ecrire un algorithme constituant un tableau, à partir de deux tableaux de même longueur préalablement saisis. Le nouveau tableau sera la somme des éléments des deux tableaux de départ.

2.3 FacProduct

Toujours à partir de deux tableaux précédemment saisis, écrire un algorithme qui calcule le 'FacProduct' des deux tableaux. Pour calculer le 'FacProduct', il faut multiplier chaque élément du tableau1 par chaque élément du tableau2, et additionner le tout.

2.4 Le plus grand

Ecrire un algorithme permettant à l'utilisateur de saisir un nombre déterminé de valeurs. Le programme, une fois la saisie terminée, renvoie la plus grande valeur en précisant quelle position elle occupe dans le tableau.

3 Maths

3.1 Le temps d'arrêt

On note H_n la somme $H_n = \sum_{k=1}^n \frac{1}{k}$. On admet que H tend vers $+\infty$. Écrire un algorithme qui détermine le plus petit entier n tel que (H) dépasse un réel α donné.

3.2 Petite equation

On considère l'équation $x^2 - 2y^2 = 1$, d'inconnues $x, y \in \mathbb{N}^*$ (les entiers). Écrire un algorithme permettant de déterminer toutes les solutions de cette équation pour lesquelles $y \leq 100$

3.3 Le classico classique : polynome 2

Saisir 3 entiers a, b, c et déterminer dans R les racines de l'équation

$$ax^2 + bx + c = 0$$

3.4 Le retour : polynome 2

Faite de même dans C

3.5 Factorielle

Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle. On rappelle qu'on calcul factorielle n tel que : $n! = \prod_{i=1}^n i = 1 \times 2 \times 3 \times \dots \times (n-1) \times n$.

3.6 Toujours plus de maths

Soit $f : [0, 1] \rightarrow R$ une fonction croissante. Pour $n \geq 1$ on pose :

$$U_n = \frac{1}{n} \sum_{k=0}^{n-1} f(k/n) \text{ et } V_n = \frac{1}{n} \sum_{k=1}^n f(k/n).$$

1-Démontrer que, pour tout $n \geq 1$, on a $U_n \leq \int_0^1 f(x)dx \leq V_n$. On admet que (U_n) et (V_n) convergent vers $\int_0^1 f(x)dx$.

2-Écrire un algorithme donnant une valeur approchée de $\int_0^1 e^{-x^2} dx$ à la centième près.