

Introduction à l'algorithmique

Travaux dirigés (TD)

Énoncé

Youssef EL ALLIOUI

y.elalloui@usms.ma

Série 1

NOTIONS DE BASE

Exercice 1.1.

- 1) Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

```
Variables A, B : Entier
Début
    A ← 1
    B ← A + 3
    A ← 3
Fin
```

- 2) Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?

```
Variables A, B, C : Entier
Début
    A ← 5
    B ← 3
    C ← A + B
    A ← 2
    C ← B - A
Fin
```

- 3) Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

```
Variables A, B : Entier
Début
    A ← 5
    B ← A + 4
```

```
A ← A + 1
B ← A - 4
Fin
```

- 4) Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?

```
Variables A, B, C : Entier
Début
    A ← 3
    B ← 10
    C ← A + B
    B ← A + B
    A ← C
Fin
```

- 5) Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

```
Variables A, B : Entier
Début
    A ← 5
    B ← 2
    A ← B
    B ← A
Fin
```

Questions : les deux dernières instructions permettent-elles d'échanger les deux valeurs de B et A ? Si l'on inverse les deux dernières instructions, cela change-t-il quelque chose ?

- 6) Écrire un algorithme permettant d'échanger les valeurs de deux variables A et B, et ce quel que soit leur contenu préalable.
- 7) On dispose de trois variables A, B et C. Ecrivez un algorithme transférant à B la valeur de A, à C la valeur de B et à A la valeur de C (toujours quels que soient les contenus préalables de ces variables).
- 8) Que produit l'algorithme suivant ?

```
Variables A, B, C : Caractères
Début
    A ← "423"
    B ← "12"
    C ← A + B
Fin
```

- 9) Que produit l'algorithme suivant ?

```
Variables A, B : Caractères
Début
    A ← "423"
    B ← "12"
    C ← A & B
Fin
```

Exercice 1.2.

Écrire un algorithme qui permet de déclarer deux variables A et B, saisir leurs valeurs, faire la somme et afficher le résultat ?

Exercice 1.3.

Écrire un algorithme qui permet de calculer et afficher la surface d'un cercle ?

Exercice 1.4.

Écrire un algorithme permettant d'échanger les valeurs de deux variables A et B, et ce quel que soit leur contenu préalable.

Exercice 1.5.

Écrire un algorithme qui permet de calculer et afficher le salaire brut d'un ouvrier connaissant le nombre d'heure et le tarif d'horaire ?

Exercice 1.6.

Écrire un algorithme qui fait la conversion d'une somme d'argent donnée en Euro, en une somme d'argent en DH ?

Exercice 1.7.

Écrire un programme qui lit le prix HT d'un article, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC correspondant.

Exercice 1.8.

On dispose de trois variables A, B et C. Ecrivez un algorithme transférant à B la valeur de A, à C la valeur de B et à A la valeur de C.

Exercice 1.9.

Écrire l'algorithme permettant de saisir l'abscisse *abs* d'un point A et de calculer son ordonné $f(abs)$:

$$f(x) = 2x^2 - 3x + 4$$

Exercice 1.10.

Écrire l'algorithme qui permet de saisir les paramètres d'une équation du second degré et de calculer son discriminant *delta*.

Exercice 1.11.

Écrire un programme qui demande deux nombres entiers à l'utilisateur, puis qui calcule et affiche la somme de ces nombres.

Exercice 1.12.

Le surveillant général d'un établissement scolaire souhaite qu'on lui écrit un programme qui calcule, pour chaque élève, la moyenne des notes des cinq matières. Ces matières sont avec leurs coefficients :

Matière	Coefficient
Math	8
Physique	6
Français	4
Anglais	2
Philosophie	2

Exercice 1.13.

Écrire un programme qui demande à l'utilisateur de taper la largeur et la longueur d'un champ et qui en affiche le périmètre et la surface.

Exercice 1.14.

Écrire un programme qui demande à l'utilisateur de taper 5 entiers et qui affiche leur moyenne. Le programme ne devra utiliser que 2 variables.

Exercice 1.15.

Écrire un programme qui demande à l'utilisateur de taper le prix *HT* d'un kilo de tomates, le nombre de kilos de tomates achetés, le taux de *TVA* (Exemple 10%, 20%, ...). Le programme affiche alors le prix *TTC* des marchandises.

BROUILLON .. NE PAS DIFFUSER

Série 2

LES STRUCTURES ALTERNATIVES

Exercice 2. 1.

Écrire l'algorithme qui permet de saisir un nombre puis déterminer s'il appartient à un intervalle donné, sachant que les extrémités de l'intervalle sont fixées par l'utilisateur.

Exercice 2. 2.

Écrire l'algorithme qui permet de saisir le jour, le mois et l'année d'une date (Mois : numéro du mois), et de déterminer si elle est correcte ou non, et ou est l'erreur. on suppose que :

- L'année doit être entre 1900 et 2017
- Le mois entre 1 et 12
- Le jour est entre 1 et 30

Exercice 2. 3.

Écrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif. Attention toutefois : on ne doit pas calculer le produit des deux nombres.

Exercice 2. 4.

Écrire l'algorithme qui permet de calculer le montant des heures supplémentaires d'un employé, sachant le prix unitaire d'une heure selon le barème suivant :

- Les 39 premières heures sans supplément,
- De la 40^{ième} à la 44^{ième} heure sont majorées de 50%,
- De la 45^{ième} à la 49^{ième} heure sont majorées de 75%,
- De la 50^{ième} heure ou plus, sont majorées de 100%.

Exercice 2. 5.

Ecrivez un algorithme qui lira au clavier l'heure et les minutes, et il affichera l'heure qu'il sera une minute plus tard.

Par exemple, si l'utilisateur tape 21 puis 32, l'algorithme doit répondre : "Dans une minute, il sera 21 heure(s) 33".

NB : on suppose que l'utilisateur entre une heure valide. Pas besoin donc de la vérifier.

Exercice 2. 6.

Vous gérez les stocks et les commandes d'une entreprise. Vous ne pouvez pas honorer une commande si vous n'avez pas la totalité de la quantité demandée. Donc créer un algorithme

permettant d'afficher la quantité que l'entreprise doit livrer au client. Si votre stock est inférieur, vous ne pouvez livrer que la quantité que vous possédez.

Exercice 2. 7.

Écrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif ou positif (on inclut cette fois le traitement du cas où le produit peut être nul). Attention toutefois, on ne doit pas calculer le produit !

Exercice 2. 8.

Écrire un algorithme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :

- « Poussin » de 6 à 7 ans
- « Pupille » de 8 à 9 ans
- « Minime » de 10 à 11 ans
- « Cadet » après 12 ans

Exercice 2. 9.

Écrire un algorithme qui, à partir d'un nombre compris entre 1 et 7, affiche le jour correspondant ?

Exercice 2. 10.

Écrire un algorithme qui à partir d'une note affiche la mention correspondante ?

Exercice 2. 11.

Écrire un algorithme qui, à partir de trois variables a , b et c de R , calcule les solutions de l'équation de la forme suivante :

$$(E) : ax^2 + bx + c = 0 \quad a, b \text{ et } c \text{ sont des réels}$$

On prendra garde à bien tester tous les cas possibles :

- a est nul, et l'équation est en fait une équation du premier degré.
Exemple : $4x - 2 = 0$ donne une unique solution : $x = 0.5$.
- Le discriminant $\Delta = (b^2 - 4ac)$ est nul, et il n'y a qu'une seule solution, appelée racine double, au problème.
Exemple : $2x^2 + 4x + 2 = 0$ donne la solution : $x = -1$.
- Le discriminant Δ est positif, et deux solutions existent : $x_1 = (-b - \sqrt{\Delta})/2a$ et $x_2 = (-b + \sqrt{\Delta})/2a$.
Exemple : $2x^2 + x - 6 = 0$ donne $x_1 = 1.5$ et $x_2 = -2$.
- Le discriminant Δ est négatif, et il n'existe pas de solutions (réelles) au problème.
Exemple : $x^2 + 1 = 0$ n'admet pas de solution dans R .

LES STRUCTURES RÉPÉTITIVES

Exercice 3. 1.

Ecrire un algorithme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne.

Exercice 3. 2.

Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.

Exercice 3. 3.

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.

Exercice 3. 4.

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 7) :

Table de 7 :

7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
...
7 x 10 = 70

Exercice 3. 5.

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 :

$$1 + 2 + 3 + 4 + 5 = 15$$

NB : on souhaite afficher uniquement le résultat, pas la décomposition du calcul.

Exercice 3. 6.

Ecrire un programme qui demande de saisir un entier N et qui affiche N! :

$$N! = N * (N - 1) * (N - 2) * ... * 3 * 2 * 1$$

Exemple : la factorielle de 8 :

$$8! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$$

Exercice 3. 7.

Ecrire un algorithme qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres :

```
Entrez le nombre numéro 1 : 12
Entrez le nombre numéro 2 : 14
...
Entrez le nombre numéro 20 : 6
Le plus grand de ces nombres est : 14
```

Modifiez ensuite l'algorithme pour que le programme affiche de surcroît en quelle position avait été saisie ce nombre :

```
C'était le nombre numéro 2
```

Exercice 3. 8.

Réécrire l'algorithme précédent, mais cette fois-ci on ne connaît pas d'avance combien l'utilisateur souhaite saisir de nombres. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

Exercice 3. 9.

Lire la suite des prix (en euros entiers et terminée par zéro) des achats d'un client. Calculer la somme qu'il doit, lire la somme qu'il paye, et simuler la remise de la monnaie en affichant les textes "10 Euros", "5 Euros" et "1 Euro" autant de fois qu'il y a de coupures de chaque sorte à rendre.

Exercice 3. 10.

Écrire un algorithme qui permette de connaître ses chances de gagner au tiercé, quarté, quinté et autres impôts volontaires.

On demande à l'utilisateur le nombre de chevaux partants, et le nombre de chevaux joués. Les deux messages affichés devront être :

```
Dans l'ordre : une chance sur X de gagner
Dans le désordre : une chance sur Y de gagner
```

X et Y nous sont donnés par la formule suivante, si n est le nombre de chevaux partants dans l'exercice 5.6 ci-dessus) :

```
X = n ! / (n - p) !
Y = n ! / (p ! * (n - p) !)
```

NB : cet algorithme peut être écrit d'une manière simple, mais relativement peu performante. Ses performances peuvent être singulièrement augmentées par une petite astuce. Vous commencerez par écrire la manière la plus simple, puis vous identifierez le problème, et écrirez une deuxième version permettant de le résoudre.

Exercice 3. 11.

Ecrire un programme qui demande à l'utilisateur de taper un entier N et qui calcule u(N) défini par :

$$\begin{aligned}u(0) &= 3 \\ u(n + 1) &= 3.u(n) + 4\end{aligned}$$

Exercice 3. 12.

Ecrire un programme qui demande à l'utilisateur de taper un entier N et qui calcule u(N) défini par :

$$\begin{aligned}u(0) &= 1 \\ u(1) &= 1 \\ u(n + 1) &= u(n) + u(n - 1)\end{aligned}$$

Exercice 3. 13.

Ecrire un programme qui permet de faire des opérations sur un entier (valeur initiale à 0). Le programme affiche la valeur de l'entier puis affiche le menu suivant :

1. Ajouter
2. Multiplier par
3. Soustraire
4. Quitter

Le programme demande alors de taper un entier entre 1 et 4. Si l'utilisateur tape une valeur entre 1 et 3, on effectue l'opération correspondante, on affiche la nouvelle valeur de l'entier puis on réaffiche le menu et ainsi de suite jusqu'à ce qu'on tape 4. Lorsqu'on tape 4, le programme se termine.

Exercice 3. 14.

Ecrire un programme qui demande à l'utilisateur de taper des entiers strictement positifs et qui affiche leur moyenne. Lorsqu'on tape une valeur négative, le programme affiche ERREUR et demande de retaper une valeur. Lorsqu'on tape 0, cela signifie que le dernier entier a été tapé. On affiche alors la moyenne. Si le nombre d'entiers tapés est égal à 0, on n'affiche PAS DE MOYENNE.

LES TABLEAUX

Exercice 4. 1.

Écrire un programme qui demande à l'utilisateur de saisir 10 entiers stockés dans un tableau. Le programme doit afficher le nombre d'entiers supérieurs ou égaux à 10.

Exercice 4. 2.

Ecrire un programme qui demande à l'utilisateur de saisir 10 entiers stockés dans un tableau ainsi qu'un entier V . Le programme doit rechercher si V se trouve dans le tableau et afficher "V se trouve dans le tableau" ou "V ne se trouve pas dans le tableau".

Exercice 4. 3.

Ecrire un programme qui demande à l'utilisateur de saisir 10 entiers stockés dans un tableau. Le programme doit ensuite afficher l'indice du plus grand élément.

Exercice 4. 4.

Ecrire un programme qui demande à l'utilisateur de saisir 10 entiers stockés dans un tableau ainsi qu'un entier V . Le programme doit rechercher si V se trouve dans le tableau et doit supprimer la première occurrence de V en décalant d'une case vers la gauche les éléments suivants et en rajoutant un 0 à la fin du tableau. Le programme doit ensuite afficher le tableau final.

Exercice 4. 5.

Ecrire un programme qui lit la dimension N d'un tableau T du type *int*, remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Calculer et afficher ensuite la somme des éléments du tableau.

Exercice 4. 6.

Ecrire un programme qui lit la dimension N d'un tableau T du type *int*, remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Effacer ensuite toutes les occurrences de la valeur 0 dans le tableau T et tasser les éléments restants. Afficher le tableau résultant.

Exercice 4. 7.

Ecrire un programme qui lit la dimension N d'un tableau T du type *int*, remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Ranger ensuite les éléments du tableau T dans l'ordre inverse sans utiliser de tableau d'aide. Afficher le tableau résultant.

Idée : Echanger les éléments du tableau à l'aide de deux indices qui parcourent le tableau en commençant respectivement au début et à la fin du tableau et qui se rencontrent en son milieu.

Exercice 4. 8.

Ecrire un programme qui lit la dimension N d'un tableau statique T du type *int*, remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Copiez ensuite toutes les composantes strictement positives dans un deuxième tableau dynamique $TPOS$ et toutes les valeurs strictement négatives dans un troisième tableau dynamique $TNEG$.

Afficher les tableaux $TPOS$ et $TNEG$.

Exercice 4. 9.

Ecrire un programme qui lit les dimensions L et C d'un tableau T à deux dimensions du type *int*. Remplir le tableau par des valeurs entrées au clavier et afficher le tableau ainsi que la somme de tous ses éléments.

Exercice 4. 10.

Ecrire un programme qui lit les dimensions L et C d'un tableau T à deux dimensions du type *int*. Remplir le tableau par des valeurs entrées au clavier et afficher le tableau ainsi que la somme de chaque ligne et de chaque colonne en n'utilisant qu'une variable d'aide pour la somme.

Exercice 4. 11. Maximum et minimum des valeurs d'un tableau

Ecrire un programme qui détermine la plus grande et la plus petite valeur dans un tableau d'entiers A . Afficher ensuite la valeur et la position du maximum et du minimum. Si le tableau contient plusieurs maximums ou minimums, le programme retiendra la position des premiers rencontrés.

Exercice 4. 12. Insérer une valeur dans un tableau trié

Un tableau dynamique A de dimension N , contient N valeurs entières triées par ordre croissant ; Ecrire un programme permettant d'insérer une valeur VAL , donnée au clavier, dans le tableau A de manière à obtenir un tableau de $N + 1$ valeurs triées.

Exercice 4. 13. Recherche d'une valeur dans un tableau

Problème : Rechercher dans un tableau d'entiers A une valeur VAL entrée au clavier. Afficher la position de VAL si elle se trouve dans le tableau, sinon afficher un message correspondant. La valeur POS qui est utilisée pour mémoriser la position de la valeur dans le tableau, aura la valeur -1 aussi longtemps que VAL n'a pas été trouvée.

Exercice 4. 14.

Ecrire un programme qui demande à l'utilisateur de taper 10 entiers qui seront stockés dans un tableau. Le programme doit trier le tableau par ordre croissant et doit afficher le tableau.

Algorithme suggéré :

- On cherche l'indice du plus petit élément parmi les indices de 0 à 9 et on échange cet élément avec $t[0]$.
- On cherche l'indice du plus petit élément parmi les indices de 1 à 9 et on échange cet élément avec $t[1]$.
- On cherche l'indice du plus petit élément parmi les indices de 2 à 9 et on échange cet élément avec $t[2]$.
- ...
- On cherche l'indice du plus petit élément parmi les indices de 8 à 9 et on échange cet élément avec $t[8]$.

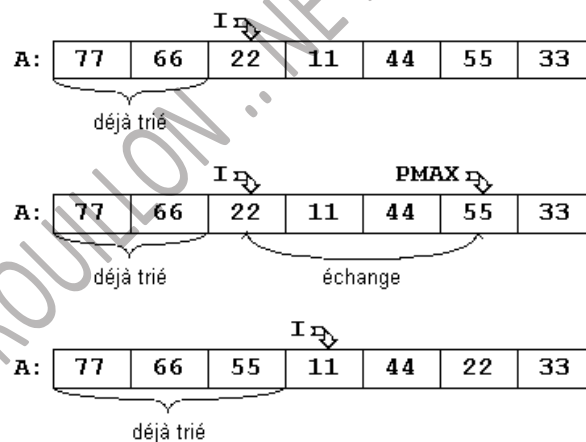
Exercice 4. 15. Fusion de deux tableaux triés

Problème: On dispose de deux tableaux A et B (de dimensions respectives N et M), triés par ordre croissant. Fusionner les éléments de A et B dans un troisième tableau FUS trié par ordre croissant.

Problème : Classer les éléments d'un tableau A par ordre décroissant.

Méthode : Parcourir le tableau de gauche à droite à l'aide de l'indice I . Pour chaque élément $A[I]$ du tableau, déterminer la position $PMAX$ du (premier) maximum à droite de $A[I]$ et échanger $A[I]$ et $A[PMAX]$.

Exemple :



Exercice 4. 16. Triangle de Pascal

Ecrire un programme qui construit le triangle de *PASCAL* de degré N et le mémorise dans une matrice carrée P de dimension $N + 1$.

Exemple : Triangle de Pascal de degré 6 :

$$\begin{array}{rcccc}
 N = 0 & & & & 1 \\
 N = 1 & & 1 & & 1 \\
 N = 2 & & 1 & 2 & 1
 \end{array}$$

$N = 3$	1	3	3	1			
$N = 4$	1	4	6	4	1		
$N = 5$	1	5	10	10	5	1	
$N = 6$	1	6	15	20	15	6	1

Méthode :

- Calculer et afficher seulement les valeurs jusqu'à la diagonale principale (incluse). Limiter le degré à entrer par l'utilisateur à 13.
- Construire le triangle ligne par ligne :
 - o Initialiser le premier élément et l'élément de la diagonale à 1.
 - o Calculer les valeurs entre les éléments initialisés de gauche à droite en utilisant la relation :

$$P_{ij} = P_{(i-1)j} + P_{(i-1)(j-1)}$$

BROUILLON .. NE PAS DIFFUSER

LES FONCTIONS

Exercice 5. 1.

Ecrire une fonction distance ayant comme paramètres 4 doubles x_a, y_a, x_b et y_b qui représentent les coordonnées de deux points A et B et qui renvoie la distance AB . Tester cette fonction.

Exercice 5. 2.

Ecrire une fonction f ayant en paramètre un entier et qui renvoie un booléen : *OUI* si l'entier est premier *NON* sinon. Tester cette fonction.

Exercice 5. 3.

Ecrire une fonction *swap* ayant en paramètres 2 entiers a et b et qui échange les contenus de a et de b . Tester cette fonction.

Exercice 5. 4.

Etablir une fonction *résoudre* permettant de calculer les solutions de l'équation :

$$(E) : ax^2 + bx + c = 0 \quad a, b \text{ et } c \text{ sont des réels}$$

Cette fonction aura comme paramètres a , b et c . On prendra garde à bien tester tous les cas possibles :

- a est nul, et l'équation est en fait une équation du premier degré. Exemple : $4x - 2 = 0$ donne une unique solution $x = 0.5$.
- Le discriminant $\Delta = (b^2 - 4 * a * c)$ est nul, et il n'y a qu'une seule solution, appelée racine double, au problème. Exemple : $2x^2 + 4x + 2 = 0$ donne $x = -1$.
- Le discriminant Δ est positif, et deux solutions existent : $x_1 = \frac{-b - \sqrt{\Delta}}{2 * a}$ et $x_2 = \frac{-b + \sqrt{\Delta}}{2 * a}$. Exemple : $2x^2 + x - 6 = 0$ donne $x_1 = 1.5$ et $x_2 = -2$.
- Le discriminant Δ est négatif, et il n'existe pas de solutions (réelles) au problème.