

FICHE DE TRAVAUX DIRIGES

ALGORITHME

UPB_2020-2021

TD N1 : Actions simples

OBJECTIFS :

- *Maîtriser les notions de base en algorithmique telles que la notion de variable, de constante et de type.*
- *Manipuler les actions simples: de lecture, d'écriture et celle d'affectation en algorithmique.*

Exercice 1:

Dire si le nom de ces variables est valide ou non valide :

prod_a		sum.of		\$total	
NewVar		Abcd		moyenne	
9ab6		DEBUT		_c3	
c123		grade1		new val	
1234		a-b		FIN_X	

Exercice 2:

Déterminer les valeurs finales des variables **A**, **B** et **C** après exécution des instructions suivantes : (les blocs d'instructions sont indépendants)

- $B \leftarrow 3$
 $A \leftarrow B$
 $C \leftarrow A$
 $B \leftarrow A * 3 + C$
- $B \leftarrow 1$
 $B \leftarrow B + 1$
 $B \leftarrow B - 3$
- $A \leftarrow 3$
 $B \leftarrow 10$
 $C \leftarrow A + B$
 $B \leftarrow A + B$
 $A \leftarrow C$
- $A \leftarrow 2$
 $B \leftarrow 4$
 $C \leftarrow 3$
 $A \leftarrow A + B + C$
 $B \leftarrow A - B - C$
 $C \leftarrow A - B - C$
 $A \leftarrow A - B - C$
- $A \leftarrow 5$
 $B \leftarrow 3$
 $C \leftarrow A + B$
 $A \leftarrow C \text{ DIV } 2$
 $B \leftarrow B + 7$
 $C \leftarrow B \text{ MOD } A$

Exercice 3:

Écrire un algorithme qui calcule et affiche l'aire d'un rectangle dont la longueur et la largeur sont égales à 17.5 et 1.4.

Exercice 4:

Déterminer et corriger les erreurs se trouvant dans les algorithmes suivants :

Algorithme calcul_aire

Début

largeur ← 15

aire ← largeur * longueur

Fin

Algorithme calcul_périmètre

Début

largeur entier

perimetre réel

perimetre ← (largeur + longueur) * 2

largeur ← 15

longueur ← 10

Fin

Algorithme calcul_moyenne

Variables note1 : réel

Début

note1 : réel

note2 : réel

moyenne : réel

lire (note2)

note1 ← 15

note2 ← 15

(note1 + note2) / 2 ← moyenne

Fin

Exercice 5:

Écrire un algorithme qui calcule le volume d'un cylindre sachant que son rayon est 5.2 et sa profondeur est de 2.8. $\text{Volume} = \text{rayon}^2 * 3.14 * \text{profondeur}$.

Exercice 6:

Écrire un algorithme qui calcule l'intérêt fixe d'un prêt dont la valeur est égale à 7000 DT et le taux est de 12.5%. $\text{Intérêt} = \text{prêt} * \text{taux}$.

Exercice 7:

Ecrire un algorithme qui lit un nombre (supposé positif) de secondes et le convertit en heures, minutes et secondes.

Exercice 8:

Écrire un algorithme qui calcule la somme de trois nombres entrés au clavier :

- En utilisant 5 variables
- En utilisant 2 variables

Exercice 9:

Ecrire un algorithme pour calculer le montant qu'un individu devra déboursier pour faire un voyage dans une automobile louée. L'algorithme tient compte du kilométrage parcouru et la durée du voyage qui sont soumises par le requérant.

Les différentes sources de dépenses sont :

- Coût de la location est de 65 D par jour.
- L'automobile louée fait 160 kilomètres pour 10 litres d'essence.
- 1 litre d'essence coûte 12.070 D.
- 0.150 D est exigé pour chaque kilomètre.

TD N2 : Les structures conditionnelles

OBJECTIFS :

Maîtriser les différentes structures conditionnelles disponibles en algorithmique.

Exercice 1:

Écrire un algorithme qui lit un entier et vérifie si cet entier est divisible par 3.

Exercice 2:

```

Algorithme abc
Variables   a,b,c, temp entier
Début
    écrire ("a=") lire (a)
    écrire ("b=")      lire (b)
    écrire ("c=") lire (c)
    si b > a alors
        temp ← a
        a ← b
        b ← temp
    fin si
    si c > a alors
        temp ← a
        a ← c
        c ← temp
    fin si
    si c > b alors
        temp ← b
        b ← c
        c ← temp
    fin si
    écrire (a,b,c)
Fin
    
```

1. Compléter ce tableau en précisant la valeur contenue dans chaque variable après l'exécution des instructions I, II et III dans les 2 cas :

	a	b	c
	1	5	10
I			
II			
III			

	a	b	c
	20	14	17
I			
II			
III			

2. Que fait cet algorithme?

Exercise 3:

Écrire un algorithme qui lit un réel et affiche sa valeur absolue.

Exercise 4:

Soient les instructions suivantes :

```
(1)  si a > 0 alors si b > 0 alors c ← a+b
                                     sinon c ← a-b
                                     finsi
```

```

    finsi
(2)  si a > 0 alors si b > 0 alors c ← a+b
      finsi
      sinon c ← a-b
    finsi

```

Quels sont les traitements réalisés dans **(1)** et **(2)** si **a** est négatif.

Exercise 5:

Écrire un algorithme qui lit un entier et affiche un message pour dire s'il est positif ou négatif ou nul.

Exercise 6:

Écrire un algorithme qui lit les paramètres d'une équation de premier degré $ax+b=0$ et affiche la solution.

Exercise 7:

Écrire un algorithme permettant de lire la valeur de la température de l'eau et d'afficher son état :

- Glace si la température est inférieure à 0, $t \leq 0$
- Eau si la température est > 0 et ≤ 100 , $0 < t \leq 100$
- Vapeur si la température est strictement supérieure à 100, $100 < t$

Exercise 8:

Écrire un algorithme qui lit un entier positif inférieur à 999 (composé de 3 chiffres au maximum) et affiche le nombre de centaines, de dizaines et d'unités.

Exercise 9:

Ecrire un algorithme qui lit trois valeurs entières (A, B et C) au clavier et qui affiche la plus grande des trois valeurs (MAX) et la plus petite d'entre elles (MIN).

Exercise 10:

Ecrire l'algorithme qui permet de saisir les trois paramètres d'une équation du second degré, et de discuter les solutions selon les valeurs de a, b et c, lorsqu'elles sont nulles ou pas.

Exercice 11:

Écrire un algorithme qui permet de saisir un numéro de couleur de l'arc-en-ciel et d'afficher la couleur correspondante

- 1-rouge
- 2-orange
- 3-jaune
- 4-vert
- 5-bleu
- 6-indigo
- 7-violet

Exercice 12:

Ce programme permet de calculer un salaire net à partir d'un salaire brut et de quelques renseignements concernant la situation de l'utilisateur. L'utilisateur devra saisir au clavier :

- son salaire brut [*montant réel*]
- sa situation :
 - a-t-il des frères et/ou des sœurs (o/n)
 - est-il boursier (o/n)

Aucun contrôle de validité de la saisie des entrées ne sera effectué. Le salaire net sera affiché à l'écran [*montant réel*]. Celui sera déterminé comme suit :

- $\text{impôt} = 0.1$
- si la personne n'a ni frère ni sœur : $\text{impôt} + 0.05$
- si la personne est boursière : $\text{impôt} - 0.05$

Le salaire net est le produit du taux d'imposition (1-impôt) et du salaire brut.

Exercice 13:

Ecrire l'algorithme qui permet de saisir deux nombres, et un opérateur et d'évaluer l'expression arithmétique correspondante.

Exercice 14:

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

1. Les 39 premières heures sans supplément
2. De 40^{ème} à 44^{ème} heure majorées de 50%
3. De 45^{ème} à 49^{ème} heure majorées de 75%
4. De 50^{ème} ou plus sont majorées de 100%

Cet algorithme lira le nombre d'heures travaillées et le prix unitaire d'une heure normale.

TD N3 : Les structures répétitives

OBJECTIFS :

Maîtriser les différentes structures répétitives disponibles en algorithmique.

Exercice 1:

1. Quelle est la différence entre la boucle tant que et la boucle répéter jusqu'à?
2. Transformer la boucle **Pour** suivante :

```
Pour i de 5 à 15 faire  
|   écrire(" le carré de ", i, " est ", i*i)  
finPour
```

- en :
- boucle **Tant que**
 - boucle **Répéter jusqu'à**

Exercice 2:

```
Algorithme exercice2  
Variables i, k, N, P : entier  
Début  
|   lire(N)  
|   i ← N  
|   P ← 1  
|   Pour i de 1 à N faire  
|   |   k ← i  
|   |   P ← P*k  
|   |   k ← k-1  
|   finfaire  
|   écrire(P)  
Fin
```

3. Donnez la trace d'exécution de cet algorithme pour **n = 4**
4. Que fait cet algorithme.

Exercice 3:

Écrire un algorithme qui lit un entier et affiche les 5 entiers suivants.

Exercice 4:

Ecrire l'algorithme qui permet de saisir les moyennes des N étudiants de la classe Informatique et de calculer la moyenne générale de la classe.

Exercice 5:

Ecrire les instructions qui permettent l'affichage de tous les nombres pairs qui existent entre 1 et 10 de trois manières différentes.

Exercice 6:

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 :

$$\begin{array}{l} 7 \times 1 = 7 \\ 7 \times 2 = 14 \\ \dots \\ 7 \times 10 = 70 \end{array}$$

Exercice 7:

Écrire un algorithme qui lit un entier et affiche son carré et ne s'arrête que si on entre la valeur 0.

Exercice 8:

Écrire un algorithme qui lit un réel x et un entier positif n et affiche x puissance n.

Exercice 9:

Écrire un algorithme qui lit un entier positif et affiche son factoriel :
 $n! = 1 * 2 * 3 * \dots * n-1 * n$ et $0! = 1$

Exercice 10:

Écrire un algorithme qui détermine si un entier N est parfait ou non. Un entier est dit parfait s'il est égal à la somme de ses diviseurs stricts (Exemple: $6 = 3 + 2 + 1$).

Exercice 11:

Écrire un algorithme qui permet de saisir un nombre entier n et d'afficher s'il est premier ou non. Un nombre premier est divisible uniquement par 1 et par lui même.

Exercice 12:

Ecrire un algorithme qui permet de saisir un nombre pair et qui détermine combien de fois ce nombre est divisible par 2.

Exemple : 8 est divisible 3 fois par 2 ($2 * 2 * 2$).

Exercice 13:

Ecrire l'algorithme qui détermine le Nième terme d'une suite définie par :

$$\begin{cases} S_0 = 2 \\ S_1 = 3 \\ S_n = S_{n-2} + (-1)^n * S_{n-1} \end{cases}$$

Exercice 14:

Écrire un algorithme qui permet de calculer le PGCD de deux nombres entiers x et y.

TD N4 : Les tableaux

OBJECTIFS :

Manipuler les différentes techniques et opérations de base sur les tableaux.

Exercice 1:

1. Que produit l'algorithme suivant ?

```
Algorithme exercice1  
Type  
  Tab=Tableau [1..5] de Entier  
Variable  
  Nb :Tab  
  i : Entier  
Début  
  | Pour i de 1 à 5 faire  
  |   Nb[i] ← i * i  
  | FinPour  
  | Pour i de 1 à 5 faire  
  |   Ecrire( Nb[i] )  
  | FinPour  
Fin
```

2. Peut-on simplifier cet algorithme avec le même résultat ?

Exercice 2:

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

Exercice 3:

Ecrivez 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.

Exemple :

Tableau 1 :

4	8	7	9	1	5	4	6
---	---	---	---	---	---	---	---

Tableau 2 :

7	6	5	2	1	3	7	4
---	---	---	---	---	---	---	---

Tableau à constituer :

11	14	12	11	2	8	11	10
----	----	----	----	---	---	----	----

Exercice 4:

Ecrivez un algorithme permettant, à l'utilisateur de saisir les notes d'une classe (max 20 étudiants) et renvoie le nombre de ces notes supérieures à la moyenne de la classe.

Exercice 5:

Pour calculer le schtroumpf, il faut multiplier chaque élément du tableau 1 par chaque élément du tableau 2, et additionner le tout.

Exemple :

Tableau 1 : 4 – 8 – 7 – 12

Tableau 2 : 3 – 6

Le Schtroumpf : $3*4 + 3*8 + 3*7 + 3*12 + 6*4 + 6*8 + 6*7 + 6*12 = 279$

Exercice 6:

Ecrire un algorithme qui lit la dimension **N** d'un tableau **T** du type entier (dimension maximale: 50 composantes) et ensuite remplit le tableau par des entiers.

Copiez ensuite toutes les composantes strictement positives dans un deuxième tableau **TP** et toutes les valeurs strictement négatives dans un troisième tableau **TN**.

Afficher les deux tableaux **TP** et **TN**.

Exercice 7:

Ecrire un algorithme qui calcule le produit scalaire de deux vecteurs d'entiers U et V (de même dimension).

Exemple:

$$\begin{array}{c} / \\ | \quad 3 \quad 2 \quad -4 \\ \backslash \end{array} * \begin{array}{c} / \\ | \quad 2 \quad -3 \quad 5 \\ \backslash \end{array} = 3*2 + 2*(-3) + (-4)*5 = -20$$

Exercice 8:

On veut écrire un algorithme qui permet de remplacer la valeur minimale d'un tableau T1 par la valeur maximale d'un autre tableau T2 et vice versa (les deux tableaux contiennent N réels, $5 \leq N \leq 50$).

Ecrire le algorithme qui permet de remplir les deux tableaux T1 et T2, de permuter le minimum de T1 et le maximum de T2 et d'afficher les résultats.

Tableaux initiaux :

T1:

2	-5	19	24	10	19	0	3
---	----	----	----	----	----	---	---

T2:

12	-6	5	-1	33	20	1	3
----	----	---	----	----	----	---	---

Résultat final :

T1:

2	33	19	24	10	19	0	3
---	----	----	----	----	----	---	---

T2:

12	-6	5	-1	-5	20	1	3
----	----	---	----	----	----	---	---

TD N 5: Les Fonctions et les procédures

Exercice 1

Écrire une fonction **Somme** qui permet de calculer la somme de n entiers saisis.

Exercice 2

Écrire une fonction **Petit** qui permet de retourner le plus petit élément d'un tableau de n entiers.

Exercice 3

Écrire une fonction **Factorielle** qui calcule la factorielle de n.

Exercice 4

Écrire une fonction **Somme1** qui permet de calculer la somme des éléments réels d'une matrice carrée comportant n lignes et n colonnes.

Exercice 5

Écrire une fonction **Puissance**, qui à partir d'un réel x et une valeur entière positive n, retourne x à la puissance n (x^n).

Exercice 6

Écrire une fonction **PGCD** qui calcule le Plus Grand Commun Diviseur de 2 entiers strictement positifs.

Exercice 7

Écrire une fonction **fonmath** qui permet de retourner $f(x,y) = xy + 5x^2 - 2y$ avec x et y deux réels.

Exercice 8

Écrire une fonction **nbreoccur** qui permet d'afficher le nombre d'occurrence d'un caractère c dans une chaîne de caractères ch.

Exercice 9

Écrire une procédure **MinMax** qui permet de chercher la plus petite et la plus grande valeur de n entiers saisis au clavier.

Exercice 10

Écrire une procédure **Suite** qui permet de chercher et d'afficher les n premiers termes de la suite V définie par :

$$V_n = \begin{cases} 4 & \text{Si } n = 0 \\ 2 V_{n-1} + 3 & \text{Si } n > 0 \end{cases}$$

Exercice 11

Écrire une procédure **Temps** qui permet de retourner le nombre d'heures, minutes et secondes pour un temps donné T sous forme d'un nombre de secondes