

## Devoir Algorithmique

### Exercice 1 :

Ecrire un algorithme qui demande le prix de fabrication et le prix de vente d'un produit et vérifiez le profit ou la perte.

Si le prix de fabrication est supérieur au prix de vente, il y a perte sinon profit.

### Exercice 2 :

Ecrire un algorithme qui demande la température en degrés Celsius dans une valeur réel, puis le convertit en Fahrenheit et affiche le résultat.

La formule de conversion est la suivante :  $Fahrenheit = (9/5) * Celsius + 32$ .

### Exercice 3 :

Ecrire un algorithme qui demande à l'utilisateur de saisir les coordonnées de deux points et qui affiche la distance entre ces deux points.

### Exercice 4 :

Ecrire un algorithme qui peut vérifier si l'année donnée par l'utilisateur est bissextile ou non.

Année bissextile c'est une année spéciale contenant un jour supplémentaire, soit un total de 366 jours dans une année. Une année est considérée comme une année bissextile si l'année est exactement divisible par 4 mais non divisible par 100. L'année est également une année bissextile si elle est exactement divisible par 400.

## Examen Algorithmique

**Exercice 1 :** On rappelle que la fonction factorielle est définie sur les entiers positifs de la façon suivante :

$$\text{factorielle}(0) = 1$$

$$\text{factorielle}(n) = n \times (n - 1) \times \dots \times 2 \times 1 \text{ si } n > 1$$

Ecrire un algorithme qui demande un nombre entier n et qui calcule la factorielle ce nombre.

**Exercice 2 :** Ecrire un algorithme qui demande un nombre entier N et qui calcule la somme des carrés des N premiers entiers impairs.

Par exemple, Si N=5 le résultat est :  $1^2 + 3^2 + 5^2 + 7^2 + 9^2 = 165$ .

Si N=7 le résultat est :  $1^2 + 3^2 + 5^2 + 7^2 + 9^2 + 11^2 + 13^2 = 455$ .

**Exercice 3 :** Quelles sont les valeurs de S et P après l'exécution après exécution des instructions suivantes ?

Variables i : Entier

S ← 0

P ← 1

**Pour** i allant de 1 à 5

S ← S + i

**Finpour**

**Pour** i allant de 1 à 5

P ← P \* i

**FinPour**

Ecrire ("S =", S, ", P=", P)

**Exercice 4 :**

Ecrire un algorithme qui demande à l'utilisateur de saisir les notes sur 20 des élèves d'une classe. Le nombre d'élèves est choisi au début du programme. Puis le programme affiche la somme, la plus grande valeur, la plus petite, et enfin, la moyenne.



## Devoir Algorithmique



### Exercice 1 :

On considère l'algorithme ci-dessous :

#### Début

```
Variables n,y : Entier
Ecrire ("Entrez un nombre n ")
Lire(n)
Si (n%2=0)
    y ← n/2
Sinon
    y ← 3*n+1
FinSi
```

Ecrire ("Le Valeur de y est ", y)

#### Fin

Faire fonctionner l'algorithme précédent et résume les résultats obtenus à chaque étape dans le tableau ci-dessous.

n	0	1	4	9	14	27
y						

### Exercice 2 :

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 toutefois : on ne doit pas calculer le produit des deux nombres.

### Exercice 3 :

On considère la fonction  $f$  définie sur  $\mathbb{R}$  par  $f(x)=x^2$  si  $x \leq -1$ ,  $f(x)=2x$  si  $-1 < x < 3$  et  $f(x)=1/x$  si  $x \geq 3$ .

Écrire un algorithme qui demande un nombre réel et qui calcule et affiche son image par  $f$ .

#### Exercice 4 :

Écrire un algorithme qui demande un nombre entier et qui calcule la valeur du  $N^{\text{ième}}$  terme ( $N < 100$ ) de la suite  $U_n$  définie par :

$$U_0=1, U_{n+1}=3U_n+1$$





## Examen Algorithmique

### Exercice 1 :

- Montrer l'exécution de l'algorithme mystère ci-contre.
- Donner la valeur finale de s après l'exécution de l'algorithme.



#### Algorithme Mystère

Variables i, j, s : Entier

Début

s ← 0 ;

Pour i allant de 1 à 4 faire

j ← 1

TantQue j ≤ 5 faire

s ← s+1 ;

j ← j+1 ;

FinTanQue

FinPour

Ecrire (s) ;

Fin

### Exercice 2 :

Ecrire un algorithme qui lit au clavier l'heure, les minutes et les secondes, et qui affiche l'heure qu'il sera une seconde plus tard. Par exemple, si l'utilisateur tape 14, puis 30, puis 45, l'algorithme doit répondre : <<Dans une seconde, il sera 14 heure(s) 30 minute(s) 46 seconde(s)>>.

### Exercice 3 :

Soit T1 un tableau de 100 valeurs ordonnées de manière croissante.

Ecrire un algorithme permettant de construire deux nouveaux tableaux ordonnés de manière croissante contenant l'un les valeurs paires, l'autre les valeurs impaires.

- Exemple :

Tableau Initial T1:

1	4	13	28	31	33	40
---	---	----	----	----	----	----

Tableau des valeurs paires T2 :

4	28	40
---	----	----

Tableau des valeurs impaires T3 :

1	13	31	33
---	----	----	----

#### Exercice 4 :

Ecrivez un algorithme permettant, à l'utilisateur de saisir les notes d'une classe en commençant par saisir le nombre d'étudiants.

L'algorithme, une fois la saisie terminée, renvoie le nombre de notes supérieures à la moyenne de la classe.



## Examen Algorithmique

### Exercice 1 :

Un nombre est n premier s'il a seulement deux diviseurs : 1 et n.

Ecrire un algorithme qui demande un nombre entier et qui teste si ce nombre est premier ou non.

**Exercice 2 :** La suite de Fibonacci ( $F_n$ ) est définie par la formule de récurrence suivante :  $F_0 = 0$ ,  $F_1 = 1$  et  $F_n = F_{n-1} + F_{n-2}$

Ecrire un algorithme qui permet de calculer le nième terme de la suite de Fibonacci, n étant fixé par l'utilisateur.

**Exercice 3 :** Quelles sont les valeurs de S et P après l'exécution après exécution des instructions suivantes ?

```
Variables i : Entier
S ← 0
P ← 1
Pour i allant de 1 à 5
    S ← S + i
Finpour
Pour i allant de 1 à 5
    P ← P * i
FinPour
Ecrire ("S =", S, ", P =", P)
```



### Exercice 4 :

Ecrire un algorithme qui demande un nombre entier et qui recherche si cet entier appartient au tableau. Au cas où la réponse est positive, l'indice de cet entier dans le tableau est affiché. S'il y a plusieurs occurrences, le dernier indice est affiché.

**N.b.:** On suppose que le tableau a été préalablement saisi.

## Devoir Algorithmique

### Exercice 1 :

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 toutefois : on ne doit pas calculer le produit des deux nombres.

### Exercice 2 :

Deux nombres sont opposés si leur somme est égale à 0. Deux nombres sont inverses si leur produit est égal à 1. Ecrire un algorithme qui demande deux nombres entiers et qui affiche si ces deux nombres sont opposés ou inverses.

### Exercice 3 :

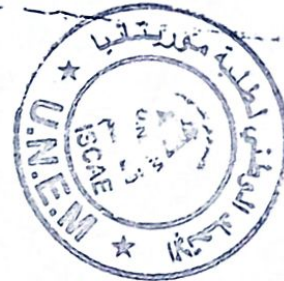
On considère l'algorithme suivant :

DEBUT  
Variables N, I, Q, S : Entier  
Lire(N)  
 $I \leftarrow N$   
 $S \leftarrow 0$

TantQue  $I > 0$  faire  
     $Q \leftarrow I \text{ Mod } 10$   
     $S \leftarrow S * 10 + Q$   
     $I \leftarrow I / 10$   
FinTantQue

Afficher (S)

FIN.



- Dérouler cet algorithme pour  $N = 52$  puis pour  $N = 123$  et donner la valeur de S pour chacune des 2 valeurs de N.
- Pour Chaque itération donner les valeurs des Q, I et S pour  $N = 52$  puis pour  $N = 123$

### Exercice 4 :

Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle.

NB : la factorielle de 8, notée  $8!$  vaut  $1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$ .



## Devoir Algorithmique

### Exercice 1 :

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

### Exercice 2 :

Ecrire un algorithme qui permet à une hôtesse de calculer le prix d'une place en fonction de l'âge du passager. Les enfants de moins de deux ans ne paient pas, ceux qui ont moins de 10 ans paient moitié prix, et les personnes de moins de 27 ans et celles de plus de 70 ans ont une réduction de 10%. L'utilisateur doit saisir le prix de base du billet et l'âge du passager. L'algorithme affiche le résultat.

### Exercice 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 4 :

On considère la suite  $(U_n)$  définie par  $U_1=4.9$  et  $U_{n+1}= 0.3 U_n$  pour tout entier naturel  $n$  non nul.

Que fait cet algorithme et donner la valeur de  $s$ ?

Variables

$k$  :Entier

$u, s$  =reel

$u \leftarrow 4.9$

$s \leftarrow 0$

Pour  $k$  de 1 à 11

$u \leftarrow 0.3 \times u$

$s \leftarrow s+u$

Fin Pour

Ecrire  $s$



## Examen Algorithmique



### Exercice 1 :

Le passage d'un étudiant inscrit en première année (L1) à la deuxième année (L2) se fait en respectant les règles suivantes :

- Si la moyenne des deux semestres S1 et S2 est supérieur ou égale à 10 alors l'étudiant est déclaré comme **admis**.
- Sinon, si l'étudiant a validé un minimum de 30 crédits avec au moins 10 crédits dans un semestre et 20 crédits dans l'autre, alors l'étudiant est déclaré comme **admis en dette**.
- Sinon, l'étudiant est déclaré comme **ajourné**.

#### Exemple :

- Si l'étudiant a 7,5 en S1 et 11,3 en S2 alors il faut voir ses crédits de S1 et S2
- S'il a 21 crédits en S1 et 11 crédits en S2 alors il est admis en dette (car il a au moins 20 crédits dans S1 et au moins 10 crédits dans S2)
- S'il a 28 crédits en S1 et 9 crédits en S2 alors il est ajourné (car il a certes au moins 20 crédits en S1 mais il n'a pas au moins 10 crédits en S2)
- S'il a 14 crédits en S1 et 15 crédits en S2 alors il est ajourné (car il n'a même pas validé le minimum de 30 crédits).

Écrire un algorithme qui demande à l'utilisateur sa moyenne en S1 et en S2 et, dans le cas échéant, ses crédits en S1 et en S2 puis lui affiche s'il est **admis**, **admis en dette** ou bien **ajourné**.

### Exercice 2 :

Quelles sont les valeurs de S et P après l'exécution après exécution des instructions suivantes ?

Variables i : Entier

$S \leftarrow 0$

$P \leftarrow 1$

Pour i allant de 1 à 5

$S \leftarrow S + i$

Finpour

Pour i allant de 1 à 5

$P \leftarrow P * i$

FinPour

Ecrire ("S =", S, ", P= ",



### Exercice 3 :

Ecrire un algorithme permettant de :

1. déclarer un tableau de taille au plus  $TMAX=100$ ,
2. saisir les éléments de ce tableau,
3. afficher les éléments de ce tableau.

### Exercice 4 :

Soient deux tableaux d'entiers T1 et T2.

Ecrire un algorithme qui permet de tester l'égalité entre les deux tableaux et affiche **VRAI** si les composants des deux tableaux correspondent position par position, et **FAUX** sinon.