

Exercice N° 1 :

Ecrire un algorithme qui demande à l'utilisateur de saisir son âge (un entier positif), puis l'algorithme lui affiche son stade de vie :

Bébé : de la naissance à 2 ans

Enfant : de 3 ans à 12 ans

Adolescent : de 13 ans à 17 ans

Adulte : de 18 ans à 69 ans

Personne âgée : à partir de 70 ans

Effectuer un contrôle de saisie sur l'âge saisi qui doit être positif.

Correction :

Algorithme age

Variables age : Entier

Début

Ecrire("Donner votre âge : ")

Lire(age)

TantQue age < 0 Faire

Ecrire("Erreur, donnez un nombre positif : ")

Lire(age)

FinTantQue

Si age <= 2 Alors

Ecrire("Bébé")

SiNon Si age <=12 Alors

Ecrire("Enfant")

SiNon Si age <=17 Alors

Ecrire("Adolescent")

SiNon Si age <=69 Alors

Ecrire("Adulte")

SiNon

Ecrire("Personne âgée")

FinSi

Fin

Exercice N° 2 :

Ecrire un algorithme qui demande à l'utilisateur un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne.

En cas de réponse supérieure à 20, on fera apparaître le message : « Plus petit ! », et inversement « Plus grand ! » si le nombre est inférieur à 10.

Correction :

Algorithme nombre

Variables n : Entier

Début

Ecrire("Donner un nombre : ")

Lire(n)

TantQue n < 10 ou n > 20 Faire

Si n < 10 Alors

Ecrire("Plus grand !")

SiNon

Ecrire("Plus petit !")

FinSi

Lire(n)

FinTantQue

Fin

Exercice N° 3 :

Ecrire un algorithme qui demande à l'utilisateur de choisir la classe de son siège pour une réservation à l'avion pour un vol de Rabat à Paris.

L'algorithme affiche le menu suivant à l'utilisateur :

Choisissez la classe :

P ---- > Première Classe (1200 Dhs)

E ---- > Classe Economique (300 Dhs)

Tapez la lettre de la classe voulue :

Ensuite le système demande à l'utilisateur le poids du bagage à enregistrer, et affiche le total à payer en ajoutant les frais de réservation de 20 Dhs et le montant à payer pour le bagage sachant que le tarif pour 1 Kg est 12 Dhs.

Effectuer un contrôle de saisie pour chaque donnée lue.

Correction :

Exercice N° 4 :

Ecrivez un algorithme qui affiche le mot « Informatique » 40 fois.

Correction :

Algorithme info

Variables i : Entier

Début

 i ← 1

 TantQue i <= 40 Faire

 Ecrire("Informatique")

 i ← i+1

 FinTantQue

Fin

Exercice N° 5 :

Ecrire un algorithme qui affiche les nombres 1 jusqu'à 40.

Correction :

Algorithme compter

Variables i : Entier

Début

 i ← 1

 TantQue i <= 40 Faire

 Ecrire(i)

 i ← i+1

 FinTantQue

Fin

Exercice N° 6 :

Ecrire un algorithme qui demande un entier positif, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, l'algorithme doit calculer :

$$1 + 2 + 3 + 4 + 5 = 15 \quad (\sum_{i=1}^5 i = 15)$$

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

Correction :

Algorithme somme

Variables i , s, n : Entier

Début

 Ecrire("Donner un entier : ")

 Lire(n)

 i ← 1

 s ← 0

```

    TantQue i <= n Faire
        s ← s + i
        i ← i+1
    FinTantQue
    Ecrire("La somme est : ", s)

```

Fin

Exercice N° 7 :

Ecrire un algorithme qui calcule et affiche le factoriel d'un entier positif entré par l'utilisateur.

Correction :

Algorithme factoriel

Variables i, f : Entier

Début

```

    Ecrire("Donner un entier : ")
    Lire(n)
    i ← 1
    f ← 1
    TantQue i <= n Faire
        f ← f * i
        i ← i+1
    FinTantQue
    Ecrire(n, " != ", f)

```

Fin

Exercice N° 8 :

Ecrire un algorithme qui calcule la somme de n entiers entrés par l'utilisateur.

Correction :

Algorithme somme

Variables i, s, n, a : Entier

Début

```

    Ecrire("Donner le nombre des valeurs : ")
    Lire(n)
    i ← 1
    s ← 0
    TantQue i <= n Faire
        Ecrire("Donner le nombre numéro ", i, " : ")
        Lire(a)
        s ← s + a
        i ← i+1
    FinTantQue
    Ecrire("La somme est : ", s)

```

Fin

Exercice N° 9 :

Ecrivez un algorithme qui calcule le reste de la division entière d'un entier positif A sur un entier strictement positif B entrés par l'utilisateur sans utiliser les deux opérateurs « div :division et mod : modulo ».

Correction :

Algorithme division

Variables A, B, r, q : Entiers

Début

```

    Ecrire("Donner les nombres A et B :")
    Lire(A, B)
    r ← A
    q ← 0
    TantQue r >= B Faire
        r ← r - B

```

```

    q ← q+1
FinTantque

Ecrire(A, " mod ", B, "=", r)
Ecrire(A, " div ", B, "=", q)
Fin

```

Exercice N° 10 :

Ecrivez un algorithme qui lit un entier strictement positif et affiche la valeur de la plus grande puissance de 2 qui soit inférieure ou égale à ce nombre.

Exemples : L'utilisateur a entré 10 → Résultat : 3 car ($2^3=8$)
 L'utilisateur a entré 73 → Résultat : 6 car ($2^6 = 64$)

Correction :

Algorithme puissance

Variables n, i : Entiers

Début

```

Ecrire("Donner un nombre :")
Lire(n)
i ← 0
TantQue  $2^i \leq n$  Faire
    i ← i+1
FinTantque
Ecrire("La plus grande puissance est : ", i-1)

```

Fin

Exercice N° 11 :

Ecrire un algorithme qui calcule le PGCD (Plus Grand Diviseur Commun) de deux nombres entiers strictement positif entrés par l'utilisateur.

Correction :

Algorithme pgcd

Variables A, B, min, i, pgcd : Entiers

Début

```

Ecrire("Donner le nombre A:")
Lire(A)
Ecrire("Donner le nombre B:")
Lire(B)
min ← A
si A > B alors
    min ← B
FinSi

i ← 1
TantQue i <= min Faire
    si A mod i=0 et B mod i =0 alors
        pgcd ← i
    FinSi
    i ← i+1
FinTantque
Ecrire("Le PGCD est : ", pgcd)

```

Fin

Algorithme pgcd_euclide

Variables A, B, R : Entiers

Début

```

Ecrire("Donner le nombre A:")
Lire(A)
Ecrire("Donner le nombre B:")
Lire(B)

```

```

TantQue B!=0 Faire

```

```

R ← A mod B
A ← B
B ← R
FinTantque
Ecrire("Le PGCD est : ", A)

```

Fin

Exercice N° 12 :

Afficher la table de conversion entre les degrés Fahrenheit et Celsius de 250 à -20 degré F par pallier de 10 degrés. On passe de x degré F au degré C en calculant $(5/9 x - 160/9)$.

Exercice N° 13 :

Ecrire un algorithme qui permet de calculer et d'afficher le nombre d'occurrences d'un chiffre ($0 \leq \text{chiffre} < 10$) dans un nombre positif.

Exemples :

L'occurrence du chiffre 7 dans le nombre 778 est 2.

L'occurrence du chiffre 8 dans le nombre 20681 est 1.

L'occurrence du chiffre 5 dans le nombre 2771 est 0.

Correction :

Algorithme occurrences

Variables n, a, u, c : Entiers

Début

```

Ecrire("Donner un nombre :")
Lire(n)
Ecrire("Donner le chiffre :")
Lire(a)

```

c ← 0

TantQue n!=0 Faire

```

u ← n mod 10
n ← n div 10
si u=a alors
    c ← c+1

```

FinSi

FinTantque

Ecrire("Le nombre d'occurrences est : ", c)

Fin

Exercice N° 14 :

Ecrire un algorithme lit un nombre entier positif et le convertit en binaire.

Correction :

Algorithme binaire

Variables n, r, q, s, i : Entiers

Début

```

Ecrire("Donner un nombre :")
Lire(n)
s ← 0
i ← 0
TantQue n!=0 faire
    r ← n mod 2
    s ← s + r*10^i
    n ← n div 2
    i ← i+1
FinTantQue
Ecrire(s)

```

Fin

Exercice N° 15 :

Écrire un algorithme qui saisit un entier et qui l'affiche à l'envers. Par exemple, l'utilisateur saisit 123456 et l'algorithme affiche 654321. Pour cela il faudra utiliser la division et le modulo.

Rappel : $153\%10 = 3$ et $153/10 = 15$

Correction :

Algorithme envers

Variables n, s : Entiers

Début

Ecrire("Donner un nombre :")

Lire(n)

$s \leftarrow 0$

TantQue $n \neq 0$ Faire

$u \leftarrow n \bmod 10$

$s \leftarrow s*10 + u$

$n \leftarrow n \div 10$

FinTantque

Ecrire("Le nombre à l'envers est : ", s)

Fin

Exercice N° 16 :

Un nombre narcissique (ou nombre d'Armstrong de première espèce est un entier naturel n non nul qui est égal à la somme des puissances p -ièmes de ses chiffres en base dix, où p désigne le nombre de chiffres de n :

Exemple :

$$153 = 1^3 + 5^3 + 3^3$$

$$548834 = 5^6 + 4^6 + 8^6 + 8^6 + 3^6 + 4^6$$

Écrire un algorithme qui lit un entier naturel et vérifie si c'est un nombre narcissique.

Exercice N° 17 :

Écrire un algorithme qui recherche le premier nombre entier naturel dont le carré se termine par n fois le même chiffre.

Exemple : pour $n = 2$, le résultat est 10 car 100 se termine par 2 fois le même chiffre

Exercice N° 18 :

On met en culture une population de $P \leq 2000$ bactéries. On suppose qu'elle augmente de 5 % tous les jours. On souhaite déterminer au bout de combien de temps cette population dépassera 3 000 bactéries. À l'aide d'une boucle **while**, construire un algorithme PYTHON permettant de résoudre le problème.

Exercice N° 19 :

On note H_n la somme $H_n = \sum_{k=1}^n \frac{1}{k}$. On admet que (H_n) tend vers $+\infty$.

Écrire un algorithme PYTHON qui détermine le plus petit entier n tel que H_n dépasse un réel a donné.

Exercice N° 20 :

On pose $p=0,9$, et on admet que la fonction $x \mapsto p^x - 1/x$ est croissante sur $[1, a]$ et décroissante sur $[a, +\infty[$, où $a > 1$. Écrire un algorithme permettant de déterminer pour quelle valeur de l'entier r le nombre $p^r - 1/r$ est maximal.

Exercice N° 21 :

Il est fréquent que l'on munisse les matricules, les codes de nomenclature, les numéros de compte, etc., d'une clé de contrôle qui permet de vérifier leur validité.

Cette clé est calculée selon un certain procédé à partir des autres chiffres. Elle est en général ajoutée à la fin du code.

Une technique simple pour produire une clé de contrôle consiste à multiplier chaque chiffre du code par un certain facteur (i.e. une pondération), à faire la somme des chiffres des produits obtenus, à diviser cette somme par 10 et à retrancher à 10 le reste de la division. Cette différence donne la clé de contrôle.

Exemple:

- | | | | | | | |
|---|----------------------|---|---|----|---|----|
| - matricule sans la clé : | 1 | 2 | 3 | 4 | 5 | 6 |
| - facteur de pondération: | 1 | 3 | 1 | 3 | 1 | 3 |
| - produit (chiffre * poids): | 1 | 6 | 3 | 12 | 5 | 18 |
| - somme des chiffres (produits): | 1 | 6 | 3 | 3 | 5 | 9 |
| - somme des sommes des chiffres: | 27 | | | | | |
| - le reste de la division euclidienne de la somme obtenue sur 10 : | 27 mod 10 = 7 | | | | | |
| - retrancher à 10 le reste la division précédente, le résultat est la clé recherchée : 10 - 7 = 3 | | | | | | |
| - matricule avec la clé : | 1 2 3 4 5 6 3 | | | | | |

- Ecrire un algorithme qui sécurise un matricule à l'aide d'une clé et un facteur de pondération entré par l'utilisateur.
- Ecrire un algorithme qui vérifie la validité d'un matricule saisi à l'aide de la clé de contrôle

Exercice N° 22 :

Dans cet exercice on va implémenter la méthode de conversion des nombres de la base binaire en décimal. La méthode consiste à sommer le résultat de multiplication de chaque bit par son poids, tel que le bit le plus à droite a le poids 0 et qui s'incrémente en allant vers les bits à gauche.

Exemple :

$$N=(100101)_2$$

On affecte le poids de chaque bit : 100101

Et on calcule la somme : $1x2^0 + 0x2^1 + 1x2^2 + 0x2^3 + 0x2^4 + 1x2^5$

$$N=(37)_{10}$$

Ecrire un algorithme qui lit un nombre binaire (un entier) et calcule et affiche sa valeur décimale.

Exercice N° 23 :

Le chiffrement ou cryptage, est un procédé de cryptographie grâce auquel on souhaite rendre la compréhension d'une information impossible à toute personne qui n'a pas la clé de déchiffrement.

Les méthodes de chiffrement les plus connues sont le DES, le Triple DES, l'AES, la RSA et le WEP.

Dans cet exercice on va coder une méthode très basique de chiffrement des nombres entiers positifs, dont le principe est le suivant :

Supposant qu'on a l'entier $n=2234$ à chiffrer avec la clé de chiffrement clé=9

- nombre avant chiffrement :

2 2 3 4

- à chaque chiffre on ajoute la clé et on ne considère que les unités du résultat :

1 1 2 3

- le nombre chiffré est :

1 1 2 3

- a) Donner le chiffrement du nombre 23598 avec la clé 12
- b) Déchiffrer le nombre 13342 avec la clé 8
- c) Ecrire un algorithme qui calcule le chiffrement d'un entier naturel entré par l'utilisateur avec une clé entrée aussi par cet utilisateur
- d) Ecrire un algorithme qui déchiffre un entier naturel entré par l'utilisateur avec une clé entrée aussi par cet utilisateur