EXERCICES D'ALGORITHMIQUE 15.2

Ex 1 - On lance une fléchette sur une cible électronique qui détecte les coordonnées (x ; y) du point d'impact F de la fléchette dans un repère orthonormal $(O\,;\vec{i}\,,\vec{j})$ gradué en cm. On s'intéresse à l'algorithme suivant :

```
Lire x et y
d = √(x² + y²)
Si d < 10
    Afficher « Bravo »
Sinon si d = 10
    Afficher « Limite »
Sinon
    Afficher « Raté »
FinSi
```

1) Traduire cet algorithme en langage Python et le tester.
2) Qu'affiche l'algorithme dans les cas suivants : a) $x = 4$; $y = 3$ b) $x = 10$; $y = 0$ c) $x = 9$; $y = 6$
3) La variable d désigne la distance entre deux points : Lesquels ?
4) De quelle forme est la cible et quelles sont ses dimensions ?

Ex 2 - On donne l'algorithme ci-dessous ?

```
Afficher « Entrez 3 nombres distincts »
Lire x, y et z
Si x < y
    x → m
Sinon
    y → m
FinSi
Si m > z
    z → m
FinSi
Afficher m
```

1) Traduire cet algorithme en langage Python.
2) Le tester dans le cas où :
   a) $x = 5$; $y = 7$; $z = 6$
   b) $x = 10$; $y = 0$; $z = -6$
3) D'une façon générale que fait cet algorithme ?

Ex 3 - On donne l'algorithme ci-dessous :

```
Lire a et b
Si a ⩾ 0
    Si b ⩾ 0
        Afficher « Le produit a × b est positif ou nul. »
    Sinon
        Afficher « Le produit a × b est négatif ou nul. »
    Fin si
Sinon
    Si b ⩾ 0
        Afficher « Le produit a × b est positif ou nul. »
    Sinon
        Afficher « Le produit a × b est négatif ou nul. »
    Fin si
FinSi
```

1) Traduire cet algorithme en langage Python.
   Attention, il y a une erreur qu'il faut corriger !
2) Modifier cet algorithme en traitant séparément **au tout début** le cas où le produit $a \times b$ est nul.

Ex 4 - On veut résoudre en fonction de a et de b l'équation du premier degré (E) : $a\,x + b = 0$

1) Résoudre cette équation dans le cas où a est différent de 0.
2) Si a est nul, il y a deux cas possibles selon que b est différent de 0 ou non.
   Résoudre l'équation dans chacun de ces deux cas.
3) Recopier et compléter le début d'algorithme ci-dessous qui doit permettre de résoudre (E) quelles que soient les valeurs entières de a et b.

```
a = int(input())
b = int(input())
if a == 0:
    if b == 0:
        print("S = R")
    else ...
```

Ex 5 - Écrire un script Python qui demande deux nombres, puis affiche la différence entre le plus grand et le plus petit.

Ex 6 - Un magasin propose de tirer des photos sur papier au tarif de 0,16 € la photo pour les 75 premières photos, puis 0,12 € la photo pour les photos suivantes. Écrire un script Python demandant à l'utilisateur d'entrer le nombre N de tirages photos commandés et calculant le montant à payer.

Ex 7 - Écrire un script Python qui demande trois nombres distincts puis les classe en ordre croissant.

Ex 8 - On considère l'algorithme ci-dessous :

```
1  Lire a et n
2  1 → p
3  Pour i de 1 à n
4      p×a → p
5  Fin pour
6  Afficher p
```

1) Tester cet algorithme « sur papier » pour $a = 2$ et $n = 3$
   On s'aidera d'un tableau où l'on donnera les valeurs de i et p après chaque exécution de la ligne 4
2) Même question pour $a = -3$ et $n = 5$
3) Traduire cet algorithme en langage Python et vérifier vos réponses aux questions 1 et 2.
4) Que calcule cet algorithme ?

Ex 9 - Dans l'algorithme ci-dessous, « ent(x) » donne la partie entière de x :

```
Lire n
Pour i de 1 à n
    Si ent(n/i) = n/i
        Afficher i
    Fin si
Fin pour
```

1) Tester cet algorithme « sur papier » pour n égal à 4 à l'aide d'un tableau dans lequel on précisera les valeurs des différentes variables en fin de boucle.
2) Recommencer avec n égal à 6
3) Quels sont les nombres que l'algorithme permet d'afficher ?
4) Traduire cet algorithme en langage Python et le tester.
5) Fonctionne-t-il encore si $n = 2,2$? et si $n = -4$?

Ex 10 - La somme des premiers nombres impairs.

1) Écrire un script Python qui permet de calculer la somme des n premiers nombres impairs.
2) En observant les résultats obtenus avec différentes valeurs de n, proposer une formule qui permet de calculer beaucoup plus rapidement cette somme.
3) En déduire sans calculatrice la somme $S = 1 + 3 + 5 + \ldots\ldots + 199\,999$.

Ex 11 - L'algorithme suivant est appelé algorithme de Héron :

```
Lire a
a/2 → b
Pour i de 1 à 4
    (b + a/b)/2 → b
Fin pour
Afficher b
```

1) Tester l'algorithme « sur papier » à l'aide d'un tableau pour $a = 2$, puis pour $a = 4$.
2) Traduire cet algorithme en langage Python et le tester.
3) Que semble calculer cet algorithme ? Peut-on le rendre plus précis ?

Ex 12 - La « factorielle » d'un entier naturel n est une fonction mathématique très utilisée en probabilités ainsi que dans un certain nombre de formules mathématiques. L'algorithme ci-dessous permet de calculer la factorielle d'un nombre entier n.

```
1 → f
Lire n
Pour i de 1 à n
    f × i → f
Fin pour
Afficher f
```

1) Tester cet algorithme « sur papier » pour n égal à 4 à l'aide d'un tableau dans lequel on précisera les valeurs des différentes variables au niveau du « Fin pour ».
   Compléter : factorielle de 4 = … × … × … × …
2) Traduire cet algorithme en langage Python et le tester avec n égal à 6.
   Compléter : factorielle de 6 = … × … × … × … × … × …
3) Quelle définition simple donneriez-vous de la factorielle d'un entier naturel quelconque n ?

Ex 13 - Écrire un script Python avec une boucle qui affiche les 10 premiers nombres pairs.

Ex 14 - Écrire un script Python avec une boucle qui calcule la somme des 1000 premiers entiers naturels.

Ex 15 - Soit f la fonction définie sur $[-1 ; 0]$ par : $x \longmapsto x^2 + x$. On considère alors l'algorithme ci-dessous.

```
–1 → x
x → a ; x² + x → b
Tant que x ⩽ 0
    x² + x → y
    Si y < b
        x → a ;  y → b
    Fin si
    x + 0,1 → x
Fin tant que
Afficher a et b
```

1) Traduire cet algorithme en langage Python et le tester.
2) Quelles sont les valeurs de a et b affichées en fin d'algorithme ?
3) Que représentent a et b pour la courbe Cf ?

Ex 16 - Quotient entier par soustractions successives :

```
0 → q
Lire a et b
Tant que a ⩾ b
    a – b → a
    q + 1 → q
Fin tant que
Afficher q
```

1) Tester « sur papier » l'algorithme avec $a = 7$ et $b = 2$ à l'aide d'un tableau dans lequel on précisera les valeurs des différentes variables en fin de boucle.
2) Traduire cet algorithme en langage Python et le tester.
3) Préciser par une phrase le rôle de chaque variable.
4) Modifier l'algorithme pour qu'il affiche aussi le reste de la division sans utiliser l'opérateur Python « % ».

Ex 17 - On considère l'algorithme ci-dessous :

$s = 0$; $i = 0$
Tant que $n \geqslant 0$
    Lire n
    $i = i+1$
    $s = s + n$
Fin tant que
$m = \dfrac{s}{i}$
Afficher m

1) Quelle variable a-t-on oublié d'initialiser en début d'algorithme ?
2) Corriger cette erreur et tester l'algorithme « sur papier » à l'aide d'un tableau dans lequel on précisera les valeurs des différentes variables en fin de boucle.
   On entrera comme valeurs successives de n : 5, 12, 13, 7, 8, 15, –3
3) Traduire cet algorithme en langage Python et le tester.
4) Préciser par une phrase le rôle de chaque variable.
5) Que doit-on entrer comme valeur de n pour quitter la boucle et lancer le calcul de m ?
6) A quoi peut servir cet algorithme ?
7) Réécrire l'algorithme pour qu'il affiche plutôt la plus grande des valeurs de n entrées.

Ex 18 - Les entiers compris entre deux réels.
1) Écrire un script Python qui demande deux réels positifs (en premier le plus petit, en second le plus grand) et qui affiche tous les entiers compris entre ces deux nombres. On s'aidera de la fonction ent(x) qui retourne la partie entière de x.
2) Même exercice dans le cas où les deux nombres entrés ne sont pas forcément dans l'ordre…

Ex 19 - La population de Bigcity augmente de 3% par an.
1) Par quel coefficient est multiplié chaque année cette population ?
2) Écrire un script Python qui permet de déterminer dans combien d'années elle aura doublé.

Ex 20 - Soit l'algorithme ci-dessous :

1 Variables :
2   a, b, n, x : réels
3 Début
4   Entrer n
5   a prend la valeur 0
6   b prend la valeur de n
7   Tant que $b - a > 0{,}01$
8   x prend la valeur de $\dfrac{a+b}{2}$
9   Si $x^2 \leqslant n$
10   a prend la valeur de x
11   Sinon
12   b prend la valeur de x
13   Fin Si
14   Fin Tant que
15   Afficher x
16 Fin.

1) Tester « sur papier » l'algorithme pour $n = 2$ en affichant les valeurs de toutes les variables juste après la ligne 13
2) Traduire cet algorithme en langage Python et le tester.
3) Que semble calculer cet algorithme ? (Pas de justification demandée)
4) Comment augmenter la précision de cet algorithme ?

Ex 21 - Dans le langage de votre calculatrice, écrire un algorithme qui demande un entier naturel puis affiche le nombre de chiffres de cet entier. (On écrira cet algorithme sur la copie en indentant le code)