EXERCICES D'ALGORITHMIQUE 7.1

Boucles « Pour » :

Ex 1 - 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 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) Que calcule cet algorithme ?

Ex 2 - On donne l'algorithme suivant où ent(n/i) donne la partie entière de n/i :

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

1) Tester cet algorithme pour n égal à 4 à l'aide d'un tableau dans lequel on précisera les valeurs des différentes variables au niveau du « Fin si ».
2) Recommencer avec n égal à 6 puis n égal à 9
3) Quels sont les nombres que l'algorithme permet d'afficher ?
4) Fonctionne-t-il encore si $n = 2{,}2$? et si $n = -4$?

Ex 3 - 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 à l'aide d'un tableau pour $a = 2$, puis pour $a = 4$.
2) Que semble calculer cet algorithme ? Peut-on le rendre plus précis ?

Ex 4 - 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 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 ». Quels produits simples avez vous fait pour calculer la factorielle de 4 ?
2) Recommencer avec n égal à 6. Quels produits simples avez vous fait pour calculer la factorielle de 6 ?
3) Quelle définition simple donneriez-vous de la factorielle d'un entier naturel quelconque n ?

Ex 5 - Écrire un algorithme avec une boucle qui affiche les 10 premiers nombres pairs.

Boucles « Tant que » :

Ex 6 - 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) Tester l'algorithme à l'aide d'un tableau dans lequel on précisera les valeurs des différentes variables au niveau du « Fin si ».
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 7 - On considère l'algorithme ci-dessous :

```
n = 0 ; s = 0 ; i = 0
Tant que n ⩾ 0
    Lire n
    i = i+1
    s = s + n
Fin tant que
m = s/i
Afficher m
```

1) Tester l'algorithme à l'aide d'un tableau dans lequel on précisera les valeurs des différentes variables au niveau du « Fin tant que ».
   On entrera comme valeurs successives de n : 5, 12, 13, 7, 8, 15, –3
2) Préciser par une phrase le rôle de chaque variable.
3) Que doit-on entrer comme valeur de n pour quitter la boucle et lancer le calcul de m ?
4) A quoi peut servir cet algorithme ?
5) Réécrire l'algorithme pour qu'il affiche plutôt la plus petite des valeurs de n entrées.

Ex 8 - 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 l'algorithme avec $a = 7$ et $b = 2$ à l'aide d'un tableau dans lequel on précisera les valeurs des différentes variables au niveau du « Fin tant que ».
2) Même question avec $a = 12$ et $b = 4$.
3) Préciser par une phrase le rôle de chaque variable.
4) Modifier l'algorithme pour qu'il affiche aussi le reste de la division.

Ex 9 - Soit f la fonction définie sur [–2,5 ; 1,5] par : $x \longmapsto 2x^2 + 2x + 1$.

1) En utilisant une boucle « Tant que », écrire un algorithme qui permette de compléter le tableau de valeurs ci-dessous :

x	–2,5	–2	–1,5	–1	–0,5	0	0,5	1	1,5
$f(x)$									

2) Même question en utilisant une boucle « Pour »

Ex 10 - Les entiers compris entre deux réels.

1) Écrire un algorithme 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...