

**TD Algorithmique et Programmation I**

**Fiche 2**

**Les structures de contrôle**

**EXERCICE 1**

Lire la note d'un candidat et le déclarer "ajourné" s'il a moins de 8, "convoqué à l'oral" s'il a de 8 compris à 10 non compris, ou "admis" s'il a 10 ou plus.

**EXERCICE 2**

Soient deux bornes Min et Max fixées en constantes dans le programme. Il s'agit de lire un nombre N et d'afficher l'un des trois messages : "Inferieur au Min", "Compris entre les bornes" (dans l'intervalle fermé), ou "Superieur au Max", selon la valeur de N. Écrire une séquence d'instructions conditionnelles ou un emboîtement d'alternatives.

**EXERCICE 3**

Lire trois nombres et dire s'ils sont tous différents (il n'y a en a pas au moins deux qui sont égaux), ou si deux au moins sont égaux, ou s'ils sont tous égaux.

**EXERCICE 4**

Lire l'heure (heures Hd et minutes Md) du départ d'un train et la durée du voyage (Hv et Mv : moins de 24 heures), puis calculer et écrire l'heure d'arrivée.

**EXERCICE 5**

Soit le programme suivant :

```
PROGRAM Inconnu; USES WINCRT; VAR i, n, m, p : INTEGER; BEGIN REPEAT  
WRITE(' Introduire un entier m : '); READLN(m); WRITE(' Introduire un entier p : ');  
READLN(p); WRITE(m :2,' avec ',p :2,' donne : '); FOR i := 1 TO m DO BEGIN n := 1 +  
p * (i - 1); WRITE(n, ' '); END; WRITELN; UNTIL m = 0; END.
```

1/ Ce programme est très mal présenté, recopiez-le proprement en utilisant les règles d'indentation du Pascal.

2/ Que fait ce programme ?

3/ Réécrivez le programme en ajoutant des commentaires.

**EXERCICE 6**

1/ Réécrire et effectuer l'algorithme suivant pour les triplets (a,b,c) :

a/ (2,-1,3)   b/ (-1,6,0)   c/ (7,4,3)

2/ Que réalise cet algorithme ?

1. début
2. lire (a,b,c) ; {a, b, c sont des entiers}
3. si  $a > b$
4. alors si  $a > c$
5. alors si  $b > c$
6. alors écrire(a, ' ', b, ' ', c)
7. sinon écrire(a, ' ', c, ' ', b)
8. sinon écrire(c, ' ', a, ' ', b)
9. sinon si  $a > c$
10. alors écrire(b, ' ', a, ' ', c)
11. sinon si  $b > c$
12. alors écrire(b, ' ', c, ' ', a)
13. sinon écrire(c, ' ', b, ' ', a)
14. fin.

### EXERCICE 7

Exécuter le programme suivant pour  $n=5$  puis 10. Que réalise t-il ?

```

début
lire (n) ; {n, p, i sont des entiers}
p ← 1
Pour i ← 1 à n faire p ← p * i ;
écrire('P=', p) ;
fin.
```

### EXERCICE 8

1/ Connaissant la valeur de  $n$ , calculer la valeur de  $n!$

Par définition  $n! = n \times (n - 1) \times (n - 2) \times \dots \times 2 \times 1$  pour  $n > 0$ .

$0! = 1$

On appelle factorielle de  $n$  le nombre  $n!$ .

Soient deux entiers  $x$  et  $n$  ( $n \geq 0$ ). Calculer  $x^n$  par multiplication successives sans utiliser l'opérateur "élévation à une puissance".

### EXERCICE 9

Après avoir corrigé les copies d'un examen, un professeur veut déterminer le nombre de notes supérieures à 10 et le nombre de notes supérieures à 18. Le nombre total de notes est connu.

### EXERCICE 10

Programme réalisant la facturation d'un article livré en un ou plusieurs exemplaires. On fournira en données le nombre d'articles et leur prix unitaire hors-taxe. La taux TVA sera toujours de 18.6 %. Si le montant TTC dépasse 1000 FF, on établira une remise de 5 %. On affichera les résultats sur 8 caractères avec 2 chiffres après la virgule.

## EXERCICE 11

Un entier  $N$  est dit parfait s'il est égal à la somme de ces diviseurs sauf lui même.

Exemple : 6 est un nombre parfait car  $6=1+2+3$ .

1/ donner la spécification du problème

2/ donner l'algorithme et l'organigramme qui représente la solution du problème

3/ le coder en turbo pascal

4/ Deux nombres entiers  $n$  et  $m$  sont dits amicaux si la somme des diviseurs de  $n$  ( $n$  non compris) vaut  $m$  et la somme des diviseurs de  $m$  ( $m$  non compris) vaut  $n$ .

Exemple : 220 et 284

Ecrire un algorithme qui détermine si deux nombres sont amicaux.

## EXERCICE 12

Ecrire un programme Pascal qui permet d'afficher le triangle de Pythagore ; triangle dans lequel l'insertion d'une ligne correspond au résultat de la multiplication de l'indice de la ligne avec les indices des colonnes.

Exemple : (pour  $n = 5$ )

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
```

## EXERCICE 13

Calculer le produit de deux nombres entiers positifs ou nuls par additions successives du multiplicande autant de fois que nécessaire.

$$a \times b = \underbrace{a + a + a + \dots + a}_{b \text{ fois}}$$

EXERCICE 14 *Multiplication de n nombres par additions successives*

Calculer le produit de  $n$  nombres entiers positifs ou nuls par additions successives selon le principe suivant :

$$\underbrace{a_1 \times a_2}_{P_1} \times a_3 \times \dots \times a_n$$

$$P_2 = P_1 \times a_3$$

$$\vdots$$

$$P_{n-1} = P_{n-2} \times a_n$$

## EXERCICE 15 (10 points)

Recopier le programme suivant en le corrigeant de toutes ses erreurs de syntaxe que vous mettrez en évidence à l'aide des numéros de ligne.

1. **Program** Mistère à découvrir
2. **Begin**

```
3.   var a;b;x;y;z :integer
4.     write('A='), readln('a');
5.     write('B='), readln('b');
6.     x=a;y=b;z=0;
7.     while y <= x then y := 2 * y;
8.     while y <> b then begin;
9.       y := y div 2;
10.      z := 2 * z;
11.      if y <= x then begin
12.        x := x - y;
13.        z := z + 1;
14.      end;
15.    end;
16.    writeln(z :5;x :5);
17.  end;
```

2/ Faire tourner le programme précédent en prenant A=1789 et B=31.

3/ De même en prenant A=2319 et B=-17.

Que se passe t-il ?

Comment doit on modifier ce programme ?

4/ Que fait ce programme ?