SOLUTION TD N°4 : LES STRUCTURES REPETITIVES

Exercice 01 : Calcule de la somme de N premiers entiers

Calculer la somme des entiers de 1 à N. N est un entier lu au clavier et supposé supérieur ou égale à 1.

- Ecrivez trois (3) algorithmes en utilisant les trois structures répétitives.

Solution :

Analyse :

➤

Algorithme 1 : Boucle « Pour »

```
Algorithme somme1;
Var N, S : entier ;
Début
  Ecrire("Entrez un nombre : ") ;
  Lire (N) ;

  S ← 0 ;
  Pour i de 1 à N Faire
    S ← S + i;
  Fpour
  Ecrire("La somme = ", S) ;
Fin.
```

Algorithme 2 : Boucle « TQ »

```
Algorithme somme2;
Var N, S, i : entier ;
Début
  Ecrire("Entrez un nombre : ") ;
  Lire (N) ;
  i ← 1 ;
  S ← 0 ;
  TQ i <= N Faire
    S ← S + i;
    i ← i + 1 ;
  Ftq
  Ecrire("La somme = ", S) ;
Fin.
```

Algorithme 3 : Boucle « Répéter »

```
Algorithme somme3;
Var N, S, i : entier ;
Début
  Ecrire("Entrez un nombre : ") ;
  Lire (N) ;
  i ← 1 ;
  S ← 0 ;
  Répéter
    S ← S + i;
    i ← i + 1 ;
  Jusqu'à (i > N)
  Ecrire("La somme = ", S) ;
Fin.
```

Exercice 02 : Les diviseurs d'un nombre

Ecrire un algorithme permettant de lire un entier N, plus grand que 1 et d'afficher tous ces diviseurs.

Solution :

➤ Cet exercice est corrigé durant le cours

Exercice 03 : Le maximum

Ecrire un algorithme qui demande successivement 20 nombre à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres.

Solution :

Algorithme 1 :

```
Algorithme maximum1;
Var n, max, i : entier ;
Début
  max ← -9999 ;
  Ecrire("Entrez les 20 nombres : ") ;
  Pour i de 1 à 20 Faire
    Lire(n);
    Si (max < n) Alors
      max ← n;
    Fsi
  Fpour
  Ecrire("Le max = ", max) ;
Fin.
```

Algorithme 2 :

```
Algorithme maximum2;
Var n, max, i : entier ;
Début
  Ecrire("Entrez le premier nombre : ") ;
  Lire (n) ;
  max ← n ;
  Ecrire("Entrez les nombres qui reste : ") ;
  Pour i de 1 à 19 Faire
    Lire(n);
    Si (max < n) Alors
      max ← n;
    Fsi
  Fpour
  Ecrire("Le max = ", max) ;
Fin.
```

Exercice 04 : Division par soustraction

Ecrire un algorithme permettant de diviser l'entier positif ou nul A par l'entier positif B.

Exemple : On veut div 20 par 3.

20 - 3 = 17	1	} 2 constitue le reste. Le quotient est le nombre de fois où on a pu retrancher 3 à 20 (6 fois)
17 - 3 = 14	2	
14 - 3 = 11	3	
11 - 3 = 8	4	
8 - 3 = 5	5	
5 - 3 = 2	6	

Diviser A par B consiste à trouver Q et R positifs ou nul tels que : $A = B*Q + R$ avec $R < B$.

Solution :

```
Algorithme division;
Var A, B, R, Q : entier ;
Début
  Ecrire("Donnez A et B : ") ;
  Lire (A, B) ;
  R ← A ;
  Q ← 0 ;
  TQ R >= B Faire
    R ← R - B;
    Q ← Q + 1 ;
  Ftq
  Ecrire("La quotient Q = ", Q, " le reste R = ", R) ;
Fin.
```

Exercice 05 : Calcule de la moyenne

Ecrire un algorithme qui demande à l'utilisateur une suite de nombres, représentant des mesures ne pouvant qu'être strictement positives. L'utilisateur ne compte pas le nombre de mesures à saisir, mais signale qu'il a terminé en entrant un nombre négatif ou nul.

- L'algorithme affiche alors le nombre de mesures saisies, et leur moyenne arithmétique.

Solution :

```
Algorithme mesures;
Var n, som, moy : réel ;
    nbr : entier ;
Début
  som ← 0 ; moy ← 0 ;
  nbr ← 0 ;
  Lire(n) ;
  TQ n > 0 Faire
    som ← som + n;
    nbr ← nbr + 1 ;
    Lire(n) ;
  Ftq
  moy ← som / nbr ;
  Ecrire("Le nombre d'élément est ", nbr) ;
  Ecrire("La moyenne des mesures est ", moy) ;
Fin.
```

Exercice 06 : Jamais content (Examen 2014/2015)

Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.

Solution :

```
Algorithme JamaisContent;
Var N : entier ;
Début
  Répéter
    Lire (N);
    Si (N > 20) Alors
      Ecrire ('Plus petit !');
    Fsi
    Si (N < 10) Alors
      Ecrire ('Plus grand !');
    Fsi
  Jusqu'à (N >= 10 et N <= 20)
  Ecrire ('Le donné entre 10 et 20 est : ', N);
Fin.
```

Exercice 07 : Table de multiplication (Examen 2013/2014)

Ecrire un algorithme qui demande un nombre entier N à l'utilisateur ($1 < N < 10$), et qui ensuite écrit la table de multiplication de ce nombre.

Exemple : Si $N = 4$, le résultat affiché est :

4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
...
...
4 x 10 = 40

Solution :

- Cet exercice est corrigé durant le cours

Exercice 08 : La suite de FIBONACCI

Construire l'algorithme qui nous calcule le Nième (avec $N > 2$) terme de la suite de FIBONACCI qui est définie

par :
$$\begin{cases} U_0 = U_1 = 1 \\ U_n = U_{n-1} + U_{n-2} \end{cases} \quad \text{pour tout } n \geq 2$$

Suite de FIBONACCI : 1, 1, 2, 3, 5, 8, 13, ...

Exemple : Si $N=5$ alors on affiche : 8

Solution :

```
Algorithme division;
Var U, N, i, U0, U1 : entier ;
Début
  Ecrire("Donnez N : ") ;
  Lire (N) ;
  Si (N = 0 OU N = 1) Alors
    U ← 1 ;
  Sinon
    U0 ← 1 ;
    U1 ← 1 ;
    Pour i de 2 à N Faire
      U ← U0 + U1 ;
      U0 ← U1 ;
      U1 ← U ;
    Fpour
  Fsi
  Ecrire("La suite de fibonacci N", N, " = ", U) ;
Fin.
```

Exercice 09 : Convertir un décimal en binaire

Construire l'algorithme de conversion d'un nombre entier N en binaire (base 2).

Exemple : $(29)_{10} = (11101)_2$

Solution :

```
Algorithme covert_bin_dec;
Var n, d, r, p : entier ;

Début
  Lire(n) ;
  d ← 0 ;
  p ← 1 ;
  Répéter
    r ← n MOD 2;
    n ← n DIV 2 ;
    d ← d + r*p ;
    p ← p*10 ;
  Jusqu'à (n = 0)
  Ecrire("Le résultat est ", d) ;
Fin.
```

Exercice 10 : l'approximation de Sin(x) (Examen 2013/2014)

Écrire un algorithme qui calcule l'approximation de Sin(x), tel que :

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

Avec x réel donné et n entier positif.

Solution :

```
Algorithme sinx;
Var sin, x : réel ;
    n, p, f, sens, i : entier ;
Début
  Lire(n) ;
  Lire(x) ;
  sin ← 0 ;
  p ← 1 ;
  f ← 1 ;
  sens ← 1 ;
  i ← 1 ;
  TQ (i < n) Faire
    sin ← sin + sens*p/f;
    i ← i + 1 ;
    sens ← sens*(-1) ;
    p ← p*x*x ;
    f ← f*(2*i)*(2*i + 1) ;
  Ftq
  Ecrire("Le résultat est ", sin) ;
Fin.
```