

## TP Structure des Ordinateurs et Applications

### Série de TP N°5 – Les instructions itératives : Pour, Tant-que et Répéter

**Exercice N°01 :** soit l'algorithme suivant :

<p><b>Algorithme</b> TP5_Exo_01 ;</p> <p><b>Variables</b> i, N, S : entier ;</p> <p><b>Début</b>          {-*-*- Entrées -*-*-}  <b>Écrire</b> ('Donner la valeur de N') ;  <b>Lire</b> (N) ;          {-*-*- Traitements -*-*-}          S ← 0 ;  <b>Pour</b> i ← 1 à N <b>faire</b>              S ← S + 2*i ;  <b>Fin-Pour</b> ;          {-*-*- Sorties -*-*-}  <b>Écrire</b> ('Le résultat S = ', S) ;  <b>Fin.</b> </p>	<p><b>Questions :</b></p> <ol style="list-style-type: none"> <li>1- Traduire cet algorithme en un programme PASCAL, puis compiler et exécuter le programme pour : <math>N = 4</math> ;</li> <li>2- Dérouler l'algorithme pour : <math>N = 4</math> ;</li> <li>3- Déduire l'expression de <math>S</math> en fonction de <math>N</math> ;</li> <li>4- Réécrire le programme en utilisant la boucle « <i>Tant-que</i> » ;</li> <li>5- Réécrire le programme en utilisant la boucle « <i>répéter</i> » ;</li> <li>6- Donner l'organigramme de cet algorithme ;</li> <li>7- Modifier le programme afin qu'il additionne les nombres jusqu'à ce que la somme dépasse 100 et afficher le nombre d'itérations <math>i</math> effectuées pour obtenir cette somme.</li> </ol>
---	--

**Exercice N°02 :**

Écrire un algorithme puis un programme en Pascal qui permet à l'utilisateur de saisir un entier positif  $N$ . Le programme doit ensuite afficher tous les nombres compris entre  $1$  et  $N$ , en indiquant pour chacun s'il s'agit d'un nombre pair ou un nombre impair.

**Exercice N°03 :**

Écrire un programme en Pascal qui permet de calculer l'énergie cinétique «  $Ec$  » d'un objet pour différentes vitesses. On suppose que l'objet a une masse constante  $m = 3 \text{ kg}$ .

La vitesse  $v$  varie de  $1 \text{ m/s}$  jusqu'à une valeur maximale  $N$  donnée par l'utilisateur, avec un pas de  $4 \text{ m/s}$ , c'est-à-dire:  $v = \{1, 5, 9, 13, \dots, N\}$ .

La formule de l'énergie cinétique est:  $Ec = \frac{1}{2} m v^2$

**Exercice N°04 :**

Écrire un programme en Pascal qui demande à l'utilisateur de saisir un **mot de passe**. Le mot de passe correct est **4951**. Tant que l'utilisateur ne tape pas ce mot de passe, le programme doit lui redemander la saisie et afficher : « *Mot de passe incorrect, veuillez réessayer.* ».

Lorsque l'utilisateur saisit enfin la valeur « **4951** », le programme affiche le message : « *Accès autorisé !* »



## TP Structure des Ordinateurs et Applications

### Exercices supplémentaires – Série de TP N°05

#### Exercice- sup N°01 :

Écrire un programme en Pascal qui permet de calculer la force gravitationnelle «  $F$  » exercée sur un objet près de la surface terrestre, pour différentes masses. On suppose que la gravité terrestre est constante et égale à  $g = 9,8 \text{ m/s}^2$ .

La masse  $m$  varie de  $3 \text{ kg}$  jusqu'à une valeur maximale  $N$  saisie par l'utilisateur, avec un pas de  $2 \text{ kg}$ . Ainsi  $m = \{3, 5, 7, 9, \dots, N\}$ .

La formule de la force gravitationnelle est :  $F = m \times g$

#### Exercice- sup N°02 :

Étant donné un entier strictement positif  $N$ , écrire un algorithme puis un programme en Pascal permettant d'afficher l'ensemble des diviseurs de  $N$ .

On rappelle qu'un diviseur d'un nombre  $N$  est tout entier positif  $d$  tel que  $N \bmod d = 0$ . Exemple  $N = 6$ , les diviseurs de  $N = \{1, 2, 3, 6\}$

#### Exercice- sup N°03 :

Écrire un programme en langage pascal pour calculer chacune des produits/sommes suivants :

$$P1 = 1 * (1 + 2) * (1 + 2 + 3) * \dots * (1 + 2 + \dots + N) \leftarrow \text{Compiler et exécuter le programme pour } N=8$$

$$S1 = 1! + 3! + 5! + \dots + N^{\text{ème}} \text{ termes} \leftarrow \text{Compiler et exécuter le programme pour } N=6$$

$$S2 = 1 + \frac{2!}{x} + \frac{4!}{x^2} + \frac{6!}{x^3} + \dots + N^{\text{ème}} \text{ termes} \leftarrow \text{Compiler et exécuter le programme pour } N=4$$

$$S3 = x - \frac{x^3}{2} + \frac{x^5}{3} - \frac{x^7}{4} + \dots \pm N^{\text{ème}} \text{ termes} \leftarrow \text{Compiler et exécuter le programme pour } N=5$$

#### Exercice- sup N°04 :

Écrire un algorithme / programme Pascal qui permet d'indiquer si entier strictement positif  $N$  est un nombre parfait ou non.

Un nombre parfait est un nombre qui à la somme de ses diviseurs propres (L'ensemble des diviseurs sauf lui-même). Exemple :  $N = 6$ , les diviseurs propres sont :  $1, 2 \text{ et } 3 \text{ et } 6=1+2+3$ , donc  $6$  est parfait.

#### Exercice- sup N°05 :

Écrire un algorithme / programme Pascal qui :

- Demande à l'utilisateur de saisir un entier  $N$  représentant le nombre de notes à traiter.
- Lit  $N$  notes réelles comprises entre 0 et 20 (avec vérification de la validité de chaque note).
- Calcule et affiche :
  - La moyenne des notes,
  - La plus grande note,
  - La plus petite note,
  - Le nombre de notes supérieures ou égales à 10.

À la fin, le programme doit afficher un message indiquant si la moyenne générale est satisfaisante ( $\geq 10$ ) ou non satisfaisante ( $< 10$ ).