

## Série d'exercices n° 3

**N.B :** Pour tous ces exercices, une solution modulaire est exigée

### Exercice 1 :

Ecrire un programme Pascal qui demande une phrase en entrée et qui affiche cette phrase de façon renversée. La phrase commence par une lettre et les mots sont séparés par un seul espace.

**Exemple:**

Votre phrase:

RESOLUTION DE PROBLEMES

Résultat:

PROBLEMES DE RESOLUTION

### Exercice 2 :

Ecrire un programme Pascal qui simule le jeu suivant :

- A tour de rôle, l'ordinateur et l'utilisateur choisissent un nombre qui ne peut prendre que trois valeurs : 0, 1 ou 2 ( l'instruction Pascal `N := RANDOM(3)` ; déterminera le choix de l'ordinateur)
- Si la différence entre les deux nombres choisis vaut :
  - 2, le joueur qui a proposé le plus grand nombre gagne un point.
  - 1, le joueur qui a proposé le plus petit nombre gagne un point.
  - 0, aucun point n'est marqué.
- Le jeu se termine quand un des deux joueurs( l'ordinateur ou l'utilisateur) totalise 10 points ou quand l'utilisateur introduit un nombre négatif qui indique sa volonté d'arrêter de jouer.

**N.B :**

La solution doit comporter au moins deux modules.

### Exercice 3 :

L'algorithme ci-dessous est celui d'un programme permettant de saisir un entier n (1 ≤ n ≤ 100) et de déterminer combien de fois il est divisible par 2.

```
0)Début Divisibles
1)Ecrire ("Donner un entier dans [1,100]")
  Lire (n)
2)Ecrire (n," est divisible ", Fn Nb_Divis2(n), " par 2")
3)Fin Divisibles
```

### Exemples :

11 est divisible 0 fois par 2

4 est divisible 2 fois par 2

8 est divisible 3 fois par 2

### Questions:

1. Traduire cet algorithme en Pascal.
2. Transformer la séquence 1) en une procédure Saisie, ajouter les contrôles nécessaires sur la saisie de n et en tenir compte dans le programme principal.
3. Ecrire la fonction Nb\_Divis2, permettant de retourner le nombre de fois où n est divisible

**Exercice 4 :**

Soit la somme  $S_n$  suivante :

$$S_n = 1 + 3/2^2 + 5/3^3 + 7/4^4 + \dots + (2n-1)/n^n$$

Ecrire un programme intitulé **SOMME** permettant de calculer et d'afficher la somme  $S_n$  pour un entier  $n$  positif donné en utilisant la formule ci-dessus.

**Exercice 5 :**

On veut écrire un programme Pascal permettant de lire deux mots  $ch1$  et  $ch2$  et d'afficher tous les caractères qui apparaissent dans les deux chaînes sans redondance.

Exemple :

Soit  $ch1 = \text{"Bonjour"}$  et  $ch2 = \text{"Bonbon"}$

Résultat : B ; o ; n

**Exercice 6 :**

Ecrire un programme intitulé **Predsucc** permettant de remplir un tableau  $T$  par  $N$  ( $4 < n < 30$ ) chiffres aléatoires compris entre 0 et 9 et d'afficher les éléments de  $T$  qui sont encadrés par leurs prédécesseurs et leurs successeurs.

**N.B :**

- Le premier élément sera affiché s'il est seulement succédé par son successeur.
- Le dernier élément sera affiché s'il est précédé par son prédécesseur

**Exercice 7 :**

On veut écrire un programme permettant de:

- saisir une chaîne de caractères **Ch**
- parcourir la chaîne **Ch** et afficher l'occurrence de chacun de ses caractères.

**Exercice 8 :**

Soit un tableau  $P$  de  $n$  chaîne de caractères ( $1 < n < 100$ ). Tous les éléments de  $T$  doivent être constitués uniquement de chiffres (0 ..9) et non vides.

On se propose de remplir un tableau  $Q$  de la façon suivante:

$Q[i]$  contiendra la chaîne de caractère  $P[i]$  écrite à l'envers.

**Exemple:**

Si  $P[2] = \text{"925"}$  Alors  $Q[2]$  contiendra la chaîne  $\text{"529"}$

Ecrire un programme permettant de saisir les éléments de  $P$ , de remplir puis d'afficher les éléments du tableau  $Q$ .

**Exercice 9 :**

Écrire un programme Pascal permettant de saisir les éléments d'un tableau  $T$  de  $N$  entiers ( $4 \leq N \leq 10$ ), déterminer puis afficher tous les éléments distincts de ce tableau.