

## Serie 3

### Exercice 1 :



Soit une salle de cinéma dans laquelle il reste des places de disponibles. Au guichet il y a une file d'attente de personnes qui achètent leurs billets. Nous voulons étudier le processus d'entrée de chaque personne dans la salle, lorsqu'il reste au moins une place libre et au moins une personne en attente. Soit :

**Nbpers** : la variable contenant le nombre de personnes qui attendent devant le guichet.

**Nbpl** : le nombre de places libre dans la salle.

Le raisonnement est le suivant: Tant que les valeurs de nbpers et de nbpl sont positives, on peut faire entrer une personne, la valeur de nbpers est alors diminuée de 1 et celle de nbpl aussi. Dès qu'il n'y a plus de personne en attente au guichet ou dès qu'il n'y a plus de place de libre dans la salle, on arrête le processus d'entrée des personnes dans la salle.

### Exercice 2 :

Un programme doit résoudre les quatre points suivants:

Lire un nombre quelconque de valeurs entières non nulles (la saisie des valeurs se termine justement lorsque l'on entre une valeur 0).

Afficher la plus grande des valeurs de la suite entrée.

Afficher la plus petite des valeurs de la suite entrée.

Calculer et afficher la moyenne de toutes les valeurs.

### Exercice 3 :

Le programme "Devinette" tire au hasard un nombre compris entre 0 et 10 et demande à l'utilisateur de trouver ce nombre. Pour cela, la méthode est la suivante:

Tirer au hasard un nombre compris entre 0 et 10.

Lire un nombre au clavier.

Tant que le nombre lu est différent de celui tiré au hasard:

    Lire un nombre au clavier.

    Compter le nombre de boucles effectuées

Afficher un message de réussite ainsi que le nombre de boucles pour y arriver!

### Exercice 4 :

Modifiez le programme "Devinette" de telle sorte que:

Les valeurs tirées au hasard s'étalent jusqu'à 50.

Qu'un message d'erreur s'affiche si la réponse est mauvaise.

Que le programme indique si la valeur saisie au clavier est plus grande ou plus petite que la valeur tirée au hasard Pour cette nouvelle version le programme s'appellera: "Jeu"

Enfin, un petit point de réflexion: Comment s'y prendre pour trouver la valeur en donnant un minimum de réponse?

### **Exercice 5 :**



Ressortons notre dé! On veut le lancer dix fois d'affilés. Il s'agit d'écrire un programme qui simule cette opération et affiche "Vous gagnez" si au moins deux six ont été tirés et " Perdu" dans le cas contraire, le tout en utilisant une boucle for ( ) pour la répétition.

### **Exercice 6 :**

Calculez la factorielle  $N! = 1*2*3*...(N-1)* N$  d'un entier naturel  $N$  en respectant que  $0!=1$ .

- a) Utilisez while,
- b) Utilisez for.

### **Exercice 7 :**

Calculez par multiplications successives  $X^N$  de deux entiers naturels  $X$  et  $N$  entrés au clavier.

### **Exercice 8 :**

Calculez la somme des  $N$  premiers termes de la série harmonique :

$$S=1 + 1/2 + 1/3 + ... + 1/N$$

### **Exercice 9 :**

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 7) :

Table de 7 :

$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

$$7 \times 3 = 21$$

...

$$7 \times 10 = 70$$

### **Exercice 10 :**

Trouvez un programme qui détermine la nieme valeur  $U_n$  ( $n$  étant fournie en donnée par l'utilisateur ) de la suite définie comme suit :

$$U_0 = 1$$

$$U_n = (2 * U_{n-1}) + 3$$

Si  $n$  est impair

$$U_n = (3 * U_{n-1}) / 7$$

Si  $n$  est pair

### **Exercice 11 :**

Écrire un programme qui, étant données deux variables, longueur et largeur, initialisées à des valeurs strictement positives quelconques, affiche un rectangle d'étoiles ayant pour longueur longueur étoiles et largeur largeur étoiles. Exemple :

Affichage d'un rectangle d'étoiles de **longueur 6 et largeur 3**.

```
*****
*****
*****
*****
```

### **Exercice 12 :**

En utilisant une double boucle, écrire un programme qui affiche une étoile, passe à la ligne, affiche deux étoiles, passe à la ligne, affiche trois étoiles... jusqu'à cinq étoiles afin d'obtenir ceci :

```
*
**
***
****
*****
```