

# BASES DE LA PROGRAMMATION L1 S1 INFORMATIQUE

## Feuille de TD n° 2

### Les boucles

#### EXERCICE 1

Écrire un programme qui demande à l'utilisateur de lui fournir un nombre entier positif et inférieur à 100 et ceci jusqu'à ce que la réponse soit satisfaisante. Le dialogue avec l'utilisateur se présentera ainsi:

donnez un entier positif inférieur à 100 : 453

donnez un entier positif inférieur à 100 : 25

merci pour le nombre 25

#### EXERCICE 2

Écrire un programme qui demande à l'utilisateur un nombre entre 2 et 9 et affiche la table de multiplication de celui-ci. La saisie du nombre est bloquante : tant que le nombre n'est pas compris entre 2 et 9 le programme redemande....

#### EXERCICE 3

A- Écrire un programme qui saisit une suite de valeurs entières non nulles. A la fin de la saisie, le programme doit afficher la moyenne des valeurs saisies. La saisie s'arrête quand l'utilisateur donne la valeur 0.

B- Écrire un programme qui saisit une suite de valeurs entières non nulles. A la fin de la saisie, le programme doit afficher la moyenne des valeurs saisies, la valeur minimale et la valeur maximale. Les trois valeurs doivent être calculées en utilisant une seule boucle. La saisie s'arrête quand l'utilisateur donne la valeur 0.

#### EXERCICE 4

A- Écrire un programme qui saisit une suite de valeurs entières non nulles. A la fin de la saisie, le programme doit afficher la moyenne des valeurs paires et la moyenne des valeurs impaires. La saisie s'arrête quand l'utilisateur donne la valeur 0. Attention la division par zéro est une opération interdite.

#### EXERCICE 5

A- Écrire un programme qui permet de déterminer le juste prix ( juste prix = 42). L'utilisateur saisit un nombre entier. Le programme répond : « inférieur » si le prix indiqué est supérieur à 42, il répond : « supérieur » si le prix indiqué est inférieur à 42 et il répond : « BRAVO! » s'il trouve le prix exact. Tant que le prix n'est pas le bon, l'utilisateur doit retenter sa chance en saisissant un nouveau nombre.

B- Modifiez le programme précédent afin que le juste prix soit généré aléatoirement et pas codé en dur à 42, qui sera évidemment inconnu de l'utilisateur. Pour cela on utilise **rand** et **srand** de la librairie `<cstdlib>`. On utilise aussi le temps pour initialiser le générateur des nombres aléatoires :

```
#include <cstdlib>
```

```
#include <ctime>
```

```
int main()
```

```
{
```

```
    // initialiser le générateur des nombres aléatoires
```

```
    srand (time(NULL)) ;
```

```
    // afficher 100 nombres aléatoires compris entre 0 et 9999.
```

```
    for(int i=0;i<100;i++) cout<<rand()%1000<<endl ;
```

```
}
```

En plus le programme doit, à la fin de chaque partie, afficher le le nombre de coups de la partie et proposer à l'utilisateur s'il veut continuer le jeu. Si la réponse est « o » alors le programme lance une nouvelle partie, sinon le programme affiche le nombre de coups moyen global et affiche « A bientôt » avant d'arrêter.

## **EXERCICE 6**

Écrire un programme qui permet de dessiner un rectangle d'étoiles \*. Le programme a besoin de 2 entrées : la largeur et la hauteur du rectangle (que l'utilisateur doit fournir).

## **EXERCICE 7**

Écrire un programme qui permet de dessiner X avec des \*. Le programme a besoin d'une entrée : la taille de la forme. La taille doit être un nombre entier impair compris entre 15 et 20 (inclus). Tant que le nombre saisi ne vérifie pas la condition, le programme redemande la valeur de la taille.

## **EXERCICE 8**

Écrire un programme qui calcule et affiche la factorielle d'un entier  $n > 0$  :

$$n! = n*(n-1)*(n-2)*...*1 ; \text{ par définition } 0! = 1.$$

## **EXERCICE 9**

Un nombre est dit premier s'il n'est divisible que par 1 par et lui même. Écrire un programme qui saisit un entier et qui affiche « PREMIER » s'il est premier et « PAS PREMIER » sinon. Modifier le programme pour qu'il saisisse deux entiers positifs et qu'il affiche tous les entiers premiers entre les deux.