TD 2
Deug S2
Itérations

Dans cette séance de travaux dirigés, nous abordons les points suivants :

- la boucle `for(initialisation;condition de continuation;increment)`
- la boucle `while (condition de continuation)`
- la boucle `do... while (condition de continuation)`

Itérations bornées :
boucle `for(initialisation;condition de continuation;increment)`

1. **Somme d'entiers**
   - Ecrire un programme qui calcule la somme des 10 premières valeurs entières entrées par l'utilisateur.
   - Ecrire un programme qui lit un entier positif `n` entré par l'utilisateur, et qui calcule la somme de au plus `n` entiers positifs entrés par l'utilisateur (si une valeur négative est entrée, elle est ignorée).
     Exemple :
     `Entrez le nombre maximum d'entiers a sommer ;` *5*
     `Entrez 5 valeurs :` *2 -4 3 6 -1*
     `Somme des valeurs positives : 11`

2. **Fonction factorielle**
   On définit la factorielle d'un entier n de la manière suivante :
$$n! = \begin{cases} \text{non défini} & \text{si } n < 0, \\ 1 & \text{si } n = 0, \\ 1 \times 2 \times 3 \times \cdots \times (n-1) \times n & \text{si } n > 0. \end{cases}$$
   Ecrire un programme qui affiche la factorielle d'un entier `n` entré par l'utilisateur. Le programme affiche un message d'erreur si l'utilisateur entre un entier négatif.

3. **Affichage d'un triangle**
   Ecrire un programme qui lit un entier `n` entré par l'utilisateur et affiche un triangle isocèle de base $2n - 1$ et de hauteur n. Par exemple, pour n=4 :

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

   Comment permettre à l'utilisateur de choisir le symbole utilisé ?

Itérations non bornées :
boucle `while` et `do ... while`

4. **Valeur maximale et minimale d'une suite d'entiers**
   On cherche à déterminer la valeur minimale et la valeur maximale d'une suite d'entiers positifs entrés au clavier par l'utilisateur. On ne possède aucune information sur le nombre d'éléments de cette suite.

   Ecrire un programme qui effectue la saisie d'une suite d'entiers. La saisie s'arrête lorsque un nombre négatif est entré. Le programme affichera alors la valeur minimale, ainsi que la valeur maximale de la suite d'entiers lus.

5. **Somme**
   On veut lire au clavier un entier strictement positif. Pour cela on repête la lecture si l'utilisateur entre une valeur incorrecte. Ecrire un programme qui lit un entier positif `n` entré par l'utilisateur, et qui calcule la somme de exactement `n` entiers positifs entrés par l'utilisateur (si une valeur négative est entrée, elle est ignorée).

6. **Saisie d'une suite croissante**
   On veut effectuer la saisie controlée d'une suite croissante d'entiers non nuls entrée au clavier par l'utilisateur. Le programme ne tient compte que des valeurs supérieures à la valeur précédemment acceptée. La saisie s'arrête lorsque l'utilisateur entre 0, et le programme affiche le nombre d'entiers pris en compte ainsi que leur somme. Par exemple, pour la suite (le symbole $\uparrow$ montre les valeurs prises en compte) :

$$\begin{array}{ccccccc} -1 & 2 & 4 & 2 & 4 & 5 & 0 \\ \uparrow & \uparrow & \uparrow & & & \uparrow & \end{array}$$

   Le programme affichera :

```
Nombre d'éléments : 4, somme : 10
```