

# TD-Boucles

## Exercice 1:

Écrire un programme

- qui affiche 100 fois "Bonjour" en utilisant une boucle "While" avec une variable de compte "count".
- Modifier le programme pour qu'il calcule la somme des numéros de 0 à 100.
- Que se passe-t-il si la condition d'arrêt est "(0)" ? "(1)" ?
- Que se passe-t-il si l'augmentation de "count" se fait en-dehors du corps de la boucle ?

## Exercice 2:

Écrire un programme qui demande à l'utilisateur un entier positif dans une boucle. Le programme s'arrête de demander l'entier quand l'utilisateur entre un nombre négatif. Modifier le programme pour:

- Faire la somme des numéros positifs entrés,
- Calculer le nombre de numéros positifs entrés,
- Calculer la moyenne des numéros positifs entrés.
- Retourner le numéro positif le plus grand entré.

## Exercice 3:

Écrire une boucle For qui affiche 100 fois les "Bonjours". Modifier le programme pour qu'il indexe les "Bonjours" par un entier. Nous devons obtenir en sortie écran:

```
1 Bonjour
2 Bonjour
3 Bonjour
...
100 Bonjour
```

#### **Exercice 4:**

Ecrire une boucle For qui calcule la somme des 1000 premier entiers.

#### **Exercice 5:**

Ecrire une boucle For qui affiche les numéros pairs de 0 à 1000.

#### **Exercice 6:**

À l'aide de boucles For imbriqués, écrire un program qui prend un entier N en entrée et qui affiche un rectangle de taille N\*N contenant le caractère "#".

Exemple: Si l'utilisateur entre 4 nous obtenons:

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

#### **Exercice 7:**

Ecrire un programme qui

- En entree: un numero N
- En sortie: affiche la table de multiplication de N.

#### **Exercice 8:**

- En entree: un numero N
- En sortie: imprimer un triangle rectangle et isocèle, rempli de "\*" avec N lignes.

Par exemple si l'utilisateur tape 5 en entrée, le programme affiche sur 5 lignes le triangle suivant:

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

Dans la première ligne on affiche 1 "\*",  
Dans la deuxième ligne on affiche 2 "\*"'

...

Dans la cinquième et dernière ligne on affiche 5 "\*".

### **Exercice 9:**

Écrire un programme qui fait la multiplication des entrees jusqu'a ce que l'utilisateur entre 0. (Tips: utiliser une boucle "do ... while".

### **Exercice 4:**

Ecrire un programme qui vérifiez si un mot (chaine de caractère) est un palindrome. En sortie, le programme affiche un message disant si c'est un palindrome ou pas.

Définition: Soit m un mot (ou chaine de caractère) et n un mot obtenu en prenant l'inverse du mot m. (pour construire n on prend les caractères un a un de la dernière lettre à la première lettre de m).

On dit que m est un palindrome si et seulement si  $m = n$ .

Autrement dit un mot qui est égale à son inverse est un palindrome.

Exemple:

ABCccccCBA est un palindrome.

AfdPfA n'est pas un palindrome

Idée:

Un mot M de taille t est un palindrome si et seulement si:

- la première lettre est égale à la dernière lettre ET
- la deuxième lettre est égale a la deuxième lettre en partant de la fin ET
- la troisième lettre est égale a la troisième lettre en partant de la fin ET

- ...

On va utiliser 2 indexes: le premier "indexGauche" qui partira de la première lettre et le deuxième "indexDroit" qui partira de la dernière lettre.

on fera avancer "indexGauche" et reculer "indexDroit" jusqu'à la ou les lettres du milieu. A chaque fois on comparera que l'on a bien la meme lettre sinon on a pas un palindrome

### **Exercice 5:**

Ecrire un programme déclarant une variable d'entier "var" et un pointeur sur variable d'entier "ptr".

- Stocker l adresse de "var" en mémoire dans "ptr".
- Modifier la valeur de "var" en passant par le pointeur "ptr"
- Afficher à chaque fois les valeurs et adresses manipulées.