Université de M'sila **Module : ASD (S1)**
Département des Mathématiques

Série d'exercice 3 (Les Boucles)

Exercice 1 : (TD&TP)

Algorithme S3_EXO1_1 **Var** n,i,s: **entier** **debut** **ecrire**("entrer un nombre entier") **lire**(n) s=n/2*(n+1) **ecrire** ("s=",s) **fin**	**Algorithme** S3_EXO1_2 **Var** n,i,s: **entier** **debut** **ecrire**("entrer un nombre entier") **lire**(n) *** **pour** i=1 à n **faire** s ← s+i **finpour** **ecrire** ("s=",s) **fin**

.

1. On doit remplacer *** dans l'algorithme S3_EXO1_2 par quoi ? Pourquoi ?
2. Réaliser l'algorithme S3_EXO1_1 pour n=5
3. Réaliser l'algorithme S3_EXO1_2 pour n=5
4. Que font les deux algorithmes précédents.
5. Traduire les deux algorithmes précédents en C et les exécuter par les mêmes valeurs de n (**travail TP**)

Exercice 2 : (TD & TP)

Ecrire un algorithme (Prog C) qui demande un nombre de départ, et qui calcule la somme de 1 jusqu'à ce dernier.
NB : si n=7 vaut $\sum_{i=1}^{7} = 1 + 2 + 3 + 4 + 5 + 6 + 7$
Qu'est-ce qu'on doit modifier pour calculer la factorielle.
NB : la factorielle de 7, notée 7 !, vaut
1 x 2 x 3 x 4 x 5 x 6 x 7

Exercice 3 : (TD & TP)

En utilisant la boucle pour (for), écrire un algorithme (Prog C) 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 x 1 = 7
7 x 2 = 14
7 x 3 = 21
...
7 x 10 = 70

Université de M'sila **Module : ASD (S1)**
Département des Mathématiques

Série d'exercice 3 (Les Boucles)

Exercice 4 : (TD & TP)

En utilisant la boucle pour (for), écrire un algorithme (Prog C) qui demande successivement 10 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 10 nombres et qui affiche de surcroît en quelle position avait été saisie ce nombre.

Exemple : si on a entré (3, 13, 5, 8, 10, 15, 7, 8, 1, 11) l'algorithme va répondre (max=15, pos=6)

Exercice 5 : (Travail à domicile)

Réécrire l'algorithme (Prog C) précédent (choisir la boucle convienne pour cela), mais cette fois-ci on ne connaît pas d'avance combien l'utilisateur souhaite saisir de nombres. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

Exercice 6 : (TD & TP)

Ecrire un algorithme (Prog C) qui permet de simuler les affichages d'un compte à rebours à partir d'un temps (minutes, secondes) donné.

Exercice 7 : (TD & TP)

Ecrire un algorithme (Prog C) qui saisit un entier supérieur à 10000 (faire la boucle **faire..tantque** pour la vérification) et qui l'affiche à l'envers. Par exemple, l'utilisateur saisit 123456 et le programme affiche 654321. Pour cela il faudra utiliser la division et le modulo.

Exercice 8 : (TD & TP)

Réécrire l'algorithme (Prog C) précédent, mais cette fois on doit calculer et afficher le nombre m inversé.

Exemple : n=56789, on doit calculer m=98765.

Exercice 9 : (TD & TP)

Ecrire un algorithme (Prog C) qui permet d'entrer un nombre réel x>10, puis le multiplier par 0,9 plusieurs fois jusqu'à sera inférieur à 10. L'algorithme doit afficher le dernier x et le nombre de multiplications réalisés.

Exercice 10 : (TD & TP)

Ecrire un algorithme (Prog C) qui permet d'afficher la table de multiplication de 1 à 10.

Exercice 11 : (TD & TP)

Ecrire un algorithme (Prog C) qui permet d'afficher les nombres parfaits qui sont inférieurs à n. un nombre parfait est un nombre égal à la somme de ses diviseurs. Exemple : 6= 3+2+1.