Boucles

Les boucles sont des structures algorithmiques d'itération. Elles permettent de recommencer un certain nombre de fois un même bloc d'instruction, selon des conditions.
Il existe plusieurs types de boucles[1]*, nous nous arrêterons plus particulièrement sur celles ayant une implémentation en Maple :*

- *Boucle "tant que", ou* **while**.

Tant que (condition) faire	*while (condition) do*
(bloc d'instructions)	*(bloc d'instructions)*
fin faire	*od ;*

  Tant que la condition est vraie, on recommence le bloc d'instructions.
- *Boucle "pour", ou* **for**

Pour variable de i à n par pas de x faire	*for variable from i to n by x do*
(bloc d'instructions)	*(bloc d'instructions)*
fin faire	*od ;*

  Pour une variable dont la valeur varie de i à n, par pas de x, on recommence le bloc d'instructions.
- *Boucle "répéter", ou* **do**

Itérer	*do*
(bloc d'instructions)	*(bloc d'instructions)*
sortir si (condition)	*if (condition) then break ;*
(bloc d'instructions)	*(bloc d'instructions)*
fin faire	*od ;*

- *Il convient de faire attention à plusieurs choses quand on écrit une boucle : Les paramètres d'initialisation de la boucle (quelles sont les valeurs de départ ?), les conditions (peut-on entrer dans la boucle ? En sortir ?) et l'ordre des instructions.*

Un exemple[2] *trivial sera :*
Pour i de 1 à 9 par pas de 2 faire
Ecrire(i) ;
fin faire
Ce qui affichera la suite 1 3 5 7 9.

Exercice 1.

Ecrire l'algorithme qui calcule factorielle n ($!n = 1 * 2 * 3 * ... * n$) par le biais d'une boucle for. Le chiffre n est lu au clavier, et le résultat final est affiché.

Traduire cet algorithme en Maple.

[1] cf. polycopié 1, pp. 47-51

[2] Traduction Maple avec une boucle for :> for i from 1 to 9 by 2 do print(i) ; od ;
Traduction Maple avec une boucle while :> i :=1 ; while i <=9 do print(i) ;i :=i+2 ;od ;

Exercice 2.

Ecrire un programme Maple qui calcule la somme des n premiers entiers pairs strictement positifs, la valeur de n étant saisie au clavier et le résultat affiché.

Exercice 3.

Ecrire un programe capable de déterminer jusqu'à quel entier il faut aller pour que la somme des premiers entiers positifs dépasse 1000.

Exercice 4.

Ecrire un programme Maple qui détermine la somme $S_n = x_1 + x_2 + ... + x_n$ de n nombres saisis successivement au clavier. La saisie des nombres sera arrêtée en tapant le mot 'fin'.

Exercice 5.

Ecrire un programme Maple, qui affiche le motif suivant :

o
oo
ooo
oooo
ooooo
oooo
ooo
oo
o

Exercice 6.

Soit le jeu dont les règles sont les suivantes : L'ordinateur tire un nombre au hasard[3] entre 0 et 1000. Le joueur a 10 essais pour trouver ce nombre, sachant qu'en cas d'erreur, le programme lui répond "trop grand" ou "trop petit". Si le joueur entre le bon nombre, il gagne, et si à la dixième tentative le joueur n'a toujours pas proposé le bon nombre, il a perdu.

Ecrire le programme en Maple.

[3]Faire myrand :=rand(i..n) en Maple, puis utiliser la fonction myrand() qui renvoie un nombre au hasard entre i et n.