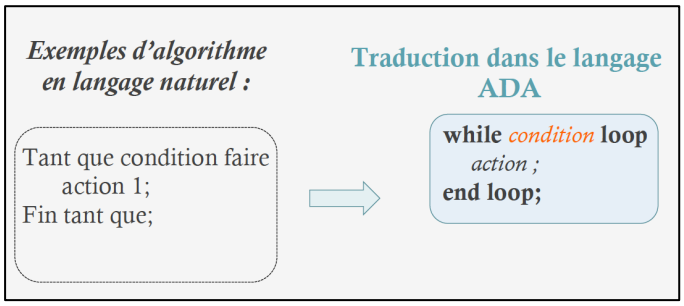
TP2 : Programmation en Ada

Objetifs : Ce TP vise à aborder les répétitions dans un programme Ada

Avant de commencer chaque TP :

1. Ouvrir un terminal et créer un dossier M1102 dans votre espace (disque H)
2. Se placer dans le dossier M1102 et créer un dossier TP2
3. Travailler dans le dossier TP2

Piqûre de rappel : Comment traduire en ada la structure de répétition « tant que… »



Exercice 1 : Boucles et autres branchements conditionnels

a) Écrire un programme ada permettant d'afficher tous les entiers compris entre 50 et 100.

b) Écrire un algorithme et un programme ada permettant d'afficher tous les entiers pairs compris entre 50 et 100. L'operateur *mod* permet de calculer la reste d'une division euclidienne.

Exercice 2 :

Nous aimerions écrire un programme qui affiche un nombre d'étoiles variable. Ce nombre d'étoiles sera fourni par l'utilisateur (on suppose sans le vérifier que ce nombre d'étoiles sera compatible avec la taille de l'écran).

Exemple : Pour une valeur saisie *nbEtoiles* égale à 8, l'algorithme affiche :

```
8 : * * * * * * * *
```

a) Écrire le programme *affichageEtoiles.adb* qui permet l'affichage précédent.
b) Testez le programme avec différentes valeurs de *NbEtoiles*.
c) Validez votre programme avec votre enseignant/e.

Nous aimerions maintenant afficher un nombre de lignes d'étoiles composées de longueurs différentes. Nous considérons que la longueur de la première ligne d'étoiles *nBEtoiles* et le nombre de lignes *nbLignes* sont saisies par l'utilisateur.

Exemple : Pour une valeur saisie nBEtoiles égale à 8 et nbLignes égale à 3, l'algorithme affiche :

8 : * * * * * * * *
7 : * * * * * * *
6 : * * * * * *

d) Modifiez le programme précédent pour obtenir l'affichage de plusieurs lignes. Remarque : Nous considérons que la valeur saisie nBEtoiles est toujours positive. Par contre, nous devrons assurer que l'affichage s'arrêt au niveau « 0 : ».

Nous aimerions permettre d'utiliser le programme précédent plusieurs fois si l'utilisateur veut recommencer. Comme vu dans le TD, nous utiliserons un marqueur de fin de lecture et tant que l'utilisateur n'aura pas tapé cette valeur comme nbLignes, le programme redemandera.

e) Écrire un nouveau programme *affichageEtoilesIteratif.adb* qui permet l'affichage itératif du programme de la question c).

Exercice 3 : Pour aller plus loin …

Nous aimerons écrire un programme pour jouer au GuessingGame dont le fonctionnement est le suivant :

- le programme déclare une variable Secret initialisée avec un nombre entre 1 et 100 ;
- il invite le joueur à saisir un entier pour deviner le nombre secret ;
- le joueur saisit un entier, le programme le lit et affiche "Votre nombre est trop petit"
- si l'entier saisi est plus grand que le nombre secret, "Votre nombre est trop grand"
- s'il est plus grand, et "C'est le bon nombre : " suivi du nombre de tentatives s'il est égal ; retour au deuxième point jusqu'à obtention de la bonne réponse.

a) Compléter le programme GuessingGame.adb qui est disponible sur Moodle. Attention ! Vous ne devez pas comprendre la génération d'un numéro aléatoire en Ada !!!