

## Feuille de TD/TP n° 2

### Structures de contrôle

#### Exercice 1 :

1. Modifier le programme de l'exercice 3 de la feuille n°1 en utilisant des `elsif` à la place des `else if`.
2. Même question en utilisant un `case`. Utiliser la clause `when others` pour signaler une erreur de saisie pour le choix de l'opération.
3. Utiliser des constantes pour définir les valeurs entières attribuées aux opérations.

**Exercice 2 :** Modifier les programmes des exercices 4 à 8 de la feuille n°1 en utilisant à la place de la boucle `while` les boucles `for`, `loop` puis l'instruction `goto`.

**Exercice 3 :** Écrire un programme utilisant une boucle et un branchement conditionnel qui affiche l'année et l'âge d'une personne depuis sa naissance jusqu'à ses 21 ans. Ajoutez une note les années d'entrée dans l'enseignement primaire, secondaire et supérieur. Par exemple :

```
En 1956, j'avais 0 an.  
En 1957, j'avais 1 an.  
En 1958, j'avais 2 ans.  
En 1959, j'avais 3 ans, et je suis entre a l'ecole primaire.  
En 1960, j'avais 4 ans.  
...
```

#### Exercice 4 :

1. Écrire un programme qui calcule la factorielle d'un entier saisi par l'utilisateur.
2. Écrire un programme qui calcule la somme des nombres impairs de 1 à un entier saisi par l'utilisateur.
3. Écrire un programme qui calcule la somme de deux entiers en n'utilisant que l'incrément de 1 comme opération.
4. Écrire un programme qui calcule la multiplication de deux entiers en n'utilisant que l'addition comme opération.
5. En déduire un programme qui calcule la multiplication de deux entiers en n'utilisant que l'incrément de 1 comme opération.

**Exercice 5 :** Soit le programme non entièrement spécifié suivant :

```
while e loop  
  i  
end loop;
```

où  $e$  représente une expression quelconque et  $i$  une suite d'instructions quelconque.

1. Proposer un programme équivalent en utilisant une boucle `loop`.
2. Proposer un programme équivalent en utilisant l'instruction `goto`.

**Exercice 6 :** Soit le programme non entièrement spécifié suivant :

```
for Index := e1 .. e2 loop
  i
end loop;
```

où  $e_1$  et  $e_2$  représentent des expressions quelconques et  $i$  une suite d'instructions quelconque.

1. Proposer un programme équivalent en utilisant une boucle `while`.
2. Proposer un programme équivalent en utilisant une boucle `loop`.
3. Proposer un programme équivalent en utilisant l'instruction `goto`.