

## Feuille d'exercices 2

### Exercice 1 :

Objectifs :

- Revoir la définition d'un sous-programme (fonction ou procédure) : paramètres, type du retour, variable locale ;
- comprendre le déroulement d'un appel de sous-programme ;
- revoir les notions de variable locale/globale, portée d'une variable dans un sous-programme, **dans un for**

L'exercice consiste à dérouler à *la main* le programme du début à la fin et expliquer ce qui se passe.

```
package main

import (
    "fmt"
)

func sousP1(p, taxe float64) float64 {
    var prix = p * (1 + taxe/100)
    fmt.Printf("p : %v \n", p)
    fmt.Printf("taxe : %v \n", taxe)
    fmt.Printf("prix : %v \n", prix)
    return prix
}

func sousP2(prix, l float64) {
    for l := 0; l < 3; l++ {
        prix = prix + 10
        fmt.Printf("l : %v prix : %v \n", l, prix)
    }
    fmt.Printf("l : %v \n", l)
    fmt.Printf("prix final : %v \n", prix)
    if prix < l {
        fmt.Println("prix final inférieur à la limite")
    } else {
        fmt.Println("prix final supérieur ou égal à la limite")
    }
}

func main() {
    var (
        prix, tva float64
    )
    prix = 12.5
    tva = 5.5
    p := sousP1(prix, tva)
    fmt.Printf("p : %v \n", p)

    sousP2(prix, 100)
}
```

**Exercice 2 :**

Point de départ : l'exercice vu en cours : *Ecrire un programme qui calcule et affiche le total à payer pour 5 articles ; l'utilisateur donne le prix (>0) de chacun des 5 articles.*

1) Reprendre l'exercice :

a) définir un jeu d'essais

b) dans repl.it :

- créer un répertoire (appelé seance4) dans le répertoire MIA0201T
- créer un repl (appelé exercice2\_1) dans lequel vous tapez le code Go
- l'exécuter en utilisant le jeu d'essais

2) A présent, c'est l'utilisateur qui indique (dans l'algorithme principal) le nombre d'articles achetés, sachant qu'il ne peut pas acheter plus de 10 articles.

a) quelles modifications devez-vous apporter à l'algorithme précédent ? Modifier l'algorithme

b) définir un jeu d'essais

c) créer un nouveau repl (exercice2\_2) avec le code Go modifié

d) l'exécuter en utilisant le jeu d'essais

**Exercice 3 :**

Reprendre l'exercice 2.2) précédent mais, cette fois-ci, calculer et afficher le prix à payer sachant que suivant le total, il bénéficie d'une réduction :

- si le total est supérieur (strictement) à 50, 20% de réduction
- si le total est compris entre 30 (compris) et 50 (compris), 15%
- si le total est compris entre 20 (compris) et 30 (non compris), 10%
- pas de réduction si le total est inférieur strictement à 20

a) quelles modifications devez-vous apporter à l'algorithme précédent ? Modifier l'algorithme

b) définir un jeu d'essais

c) créer un nouveau repl (exercice3)

d) le tester en utilisant le jeu d'essais

**Exercice 4 :**

Point de départ : l'exercice 2.2) précédent.

A présent, on ne veut plus calculer et afficher le total à payer mais le prix de l'article le plus cher.

a) concevoir l'algorithme en utilisant la méthode vue au semestre 1

b) définir un jeu d'essais

c) créer un nouveau repl (exercice4)

d) le tester en utilisant le jeu d'essais

**Exercice 5 :**

Point de départ : l'exercice 2.2) précédent.

A présent, l'utilisateur n'indique plus le nombre d'articles achetés mais donne un prix égal à 0 quand il a fini sa saisie. On veut calculer et afficher le total à payer (comme en 1 et 2)

a) concevoir l'algorithme par raffinages successifs

b) définir un jeu d'essais

c) créer un nouveau repl (exercice5)

d) le tester en utilisant le jeu d'essais

**Exercice 6 :**

Reprendre le 5) mais cette fois-ci pour calculer et afficher la moyenne des prix.

a) concevoir l'algorithme par raffinages successifs

b) définir un jeu d'essais

c) créer un nouveau repl (exercice6)

d) le tester en utilisant le jeu d'essais