Fonctions

Exercice 1

- Créez une fonction nommée `add` qui prend deux arguments de type `int` et retourne leur somme. Appelez la fonction `add` avec les arguments `5` et `3`, puis imprimez le résultat.
- Créez une fonction nommée `multiply` qui prend deux arguments de type `int` et retourne leur produit. Appelez la fonction `multiply` avec les arguments `4` et `7`, puis imprimez le résultat.
- Créez une fonction nommée `greet` qui prend un argument de type `string` (un nom) et retourne une chaîne de caractères de salutation. Appelez la fonction `greet` avec le nom `"Alice"`, puis imprimez la salutation.
- Créez une fonction nommée `isEven` qui prend un argument de type `int` et retourne `true` si le nombre est pair et `false` sinon. Appelez la fonction `isEven` avec les nombres `6` et `9`, puis imprimez les résultats.
- Créez une fonction nommée `average` qui prend un tableau dynamique (slice) de type `float64` et retourne leur moyenne. Appelez la fonction `average` avec les nombres `2.5`, `3.7` et `4.8`, puis imprimez la moyenne avec une précision de
  2.

```
package main

func main() {
    // TODO: add
    // TODO: multiply
    // TODO: greet
    // TODO: isEven
    // TODO: average
}
```

Résultat attendu:

```
8
28
Hello Alice!
6 is even: true
9 is even: false
Average: 3.67
```

Exercice 2

Créez un programme qui génère des mots de passe aléatoires avec des options personnalisables.

- L'utilisateur doit entrer la longueur souhaitée pour le mot de passe.
- L'utilisateur doit également avoir la possibilité de choisir si le mot de passe doit contenir des lettres majuscules, des lettres minuscules, des chiffres et des caractères spéciaux.
- Créez une fonction `generatePassword` qui prend en entrée la longueur du mot de passe et les options (**majuscules**, **minuscules**, **chiffres**, **caractères spéciaux**) et retourne un mot de passe aléatoire.
- Utilisez une fonction séparée `getRandomChar` pour générer un caractère aléatoire en fonction des options spécifiées.
- Utilisez cette fonction pour générer un mot de passe et affichez-le à l'écran.

Contraintes

- Utilisez au moins deux fonctions : `generatePassword` et `getRandomChar`.

- Vous devez gérer les erreurs d'entrée de l'utilisateur.
- La fonction generatePassword doit appeler la fonction getRandomChar pour générer chaque caractère du mot de passe.
- Le mot de passe doit être généré en fonction des options spécifiées par l'utilisateur (majuscules, minuscules, chiffres, caractères spéciaux).
- Le code doit être modulaire et facile à comprendre.

```
package main

func generatePassword(length int, uppercase bool, lowercase bool, numbers bool, specials bool)
string {
    // TODO: generatePassword
}

func getRandomChar(uppercase bool, lowercase bool, numbers bool, specials bool) string {
    // TODO: getRandomChar
}

func main() {
    // TODO: prompt user for input
}
```

Résultat attendu:

```
$ go run main.go

Générateur de Mot de Passe Améliorée
====================================

Entrez la longueur du mot de passe : 12

Inclure des majuscules ? (Oui/Non) : Oui
Inclure des minuscules ? (Oui/Non) : Oui
Inclure des chiffres ? (Oui/Non) : Oui
Inclure des caractères spéciaux ? (Oui/Non) : Non

Mot de passe généré : Kp7bN9A8L3j6z
```

Exercice 3

Créez un convertisseur de température qui convertit entre les échelles Celsius, Fahrenheit et Kelvin.

- Créez une fonction `convertTemperature` qui prend une température, une échelle de départ et une échelle de destination en entrée, et renvoie la température convertie.
- Créez des fonctions pour convertir les températures entre les échelles Celsius, Fahrenheit et Kelvin.
- Les échelles de température prises en charge sont Celsius, Fahrenheit et Kelvin.
- Utilisez les formules de conversion appropriées.
  - Celsius vers Fahrenheit : `T(°F) = T(°C) × 9/5 + 32`
  - Celsius vers Kelvin : `T(K) = T(°C) + 273.15`
  - Fahrenheit vers Celsius : `T(°C) = (T(°F) - 32) × 5/9`
  - Fahrenheit vers Kelvin : `T(K) = (T(°F) + 459.67)× 5/9`
  - Kelvin vers Celsius : `T(°C) = T(K) - 273.15`
  - Kelvin vers Fahrenheit : `T(°F) = T(K) × 9/5 - 459.67`

```go
package main

func convertTemperature(temperature float64, from string, to string) float64 {
    // TODO: convertTemperature
}

func main() {
    // TODO: prompt user for input
}
```

Résultat attendu:

```
go run main.go

Convertisseur de Température
---------------------------
1. Celsius vers Fahrenheit
2. Celsius vers Kelvin
3. Fahrenheit vers Celsius
4. Fahrenheit vers Kelvin
5. Kelvin vers Celsius
6. Kelvin vers Fahrenheit

Sélectionnez une option : 2
Entrez la température en degrés Celsius : 25
Température en Kelvin : 298.15
```

Exercice 4

Créez un programme qui génère des badges personnalisés pour un événement.

- Vous êtes chargé de créer des badges pour un événement et de les imprimer pour les participants.
- Créez une fonction `createBadge` qui prend en entrée le nom et le titre d'un participant, puis génère un badge au format spécifié.
- Le badge doit être formaté comme suit :

  ```
  *****************
  Nom: [Nom du Participant]
  Titre: [Titre du Participant]
  *****************
  ```

- L'utilisateur doit pouvoir entrer le nombre de participants et leurs informations.
- Utilisez la fonction createBadge pour générer les badges et affichez-les à l'écran.
- Chaque badge doit être `imprimé` avec des astérisques autour pour un effet visuel élégant.
- Utilisez la fonction `strings.Repeat` pour générer les astérisques.
- Vous pouvez aussi créer une fonction `prompt` pour demander à l'utilisateur d'entrer des informations et gérer le menu.

```go
package main

func createBadge(name string, title string) string {
    // TODO: createBadge
}

func main() {
```

```
    // TODO: prompt user for input
}
```

Résultat attendu:

```
go run main.go

Générateur de Badges pour un Événement
--------------------------------------

Combien de participants sont présents à l'événement ? : 3

Participant 1 :
Entrez le nom : Alice Johnson
Entrez le titre : Développeur Web

Participant 2 :
Entrez le nom : Bob Smith
Entrez le titre : Designer Graphique

Participant 3 :
Entrez le nom : Charlie Davis
Entrez le titre : Gestionnaire de Projet

Badges générés pour l'événement :
**********
Nom: Alice Johnson
Titre: Développeur Web
**********
**********
Nom: Bob Smith
Titre: Designer Graphique
**********
**********
Nom: Charlie Davis
Titre: Gestionnaire de Projet
**********
```

Exercice 5

Reprenons un exercice précédent, sur la gestion d'un magasin de jeux vidéo.

Proposez un code utilisant des fonctions pour simplifier le code.

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

type item struct {
    id    int
```

```
    name  string
    price int
}

type game struct {
    item
    genre string
}

func main() {
    games := []game{
        {
            item:  item{id: 1, name: "god of war", price: 50},
            genre: "action adventure",
        },
        {
            item:  item{id: 2, name: "x-com 2", price: 40},
            genre: "strategy",
        },
        {
            item:  item{id: 3, name: "minecraft", price: 20},
            genre: "sandbox",
        },
    }

    // Index les jeux par id
    gamesByID := make(map[int]game)
    for _, g := range games {
        gamesByID[g.id] = g
    }

    fmt.Printf("Le magasin propose %d jeux.\n", len(games))

    in := bufio.NewScanner(os.Stdin)
    for {
        fmt.Print(`
Commandes :
> list : liste tous les jeux
> id N : affiche le jeu d'identifiant N
> add  : ajoute un jeu
> quit : quitte le programme

`)
        fmt.Print("Votre choix : ")
        in.Scan()
        fmt.Printf("\n")

        cmd := strings.Fields(in.Text())

        if len(cmd) == 0 {
            // Pas de commande, on continue
            continue
        }

        switch cmd[0] {
        case "quit":
```

```
        fmt.Println("Au revoir !")
        return
    case "list":
        for _, g := range games {
            fmt.Printf("#%-4d: %-15q %-20s %d€\n",
                g.id,
                g.name,
                "("+g.genre+")",
                g.price,
            )
        }
    case "id":
        if len(cmd) != 2 {
            fmt.Println("ID invalide.")
            continue
        }
        id, err := strconv.Atoi(cmd[1])
        if err != nil {
            fmt.Println("ID invalide.")
            continue
        }

        g, ok := gamesByID[id]
        if !ok {
            fmt.Println("Jeu introuvable.")
            continue
        }
        fmt.Printf("#%-4d: %-15q %-20s %d€\n",
            g.id,
            g.name,
            "("+g.genre+")",
            g.price,
        )
    case "add":
        correctValue := false
        var name, genre string
        var price int
        var err error

        for !correctValue {
            fmt.Print("Nom du jeu : ")
            in.Scan()
            name = strings.TrimSpace(in.Text())
            if name == "" {
                fmt.Println("Nom invalide.")
                continue
            }
            correctValue = true
        }

        correctValue = false
        for !correctValue {
            fmt.Print("Genre du jeu : ")
            in.Scan()
            genre = strings.TrimSpace(in.Text())
            if genre == "" {
```

```
                fmt.Println("Genre invalide.")
                continue
            }
            correctValue = true
        }

        correctValue = false
        for !correctValue {
            fmt.Print("Prix du jeu : ")
            in.Scan()
            price, err = strconv.Atoi(in.Text())
            if err != nil {
                fmt.Println("Prix invalide.")
                continue
            }
            correctValue = true
        }

        newGame := game{
            item: item{
                id:    len(games) + 1,
                name:  name,
                price: price,
            },
            genre: genre,
        }

        games = append(games, newGame)
        gamesByID[newGame.id] = newGame
        fmt.Printf("\nNouveau jeu ajouté :\n#%-4d: %-15q %-20s %d€\n",
            newGame.id,
            newGame.name,
            "("+newGame.genre+")",
            newGame.price,
        )
    default:
        fmt.Println("Commande inconnue.")
    }

  }
}
```