Structures de contrôle

Exercice 1

Commençons simplement. Affichez les résultats attendus en fonction de la variable `age`.

```
package main

func main() {
    // Modifiez ceci en conséquence pour produire les résultats attendus
    // age := 10

    // Saisissez votre instruction if ici.
}
```

Résultat attendu

```
Si l'âge est supérieur à 60, affichez :
    Getting older
Si l'âge est supérieur à 30, affichez :
    Getting wiser
Si l'âge est supérieur à 20, affichez :
    Adulthood
Si l'âge est supérieur à 10, affichez :
    Young blood
Sinon, affichez :
    Booting up
```

Exercice 2

Pouvez-vous simplifier l'instruction `if` dans le code ci-dessous ?

Lorsque :

- `isSphere` est vrai (true)
- Le rayon (`radius`) est égal ou supérieur à 200

Il devrait afficher "C'est une grosse sphère."

Sinon, il devrait afficher "Je ne sais pas."

```
package main

import "fmt"

func main() {
    // NE MODIFIEZ PAS CE CODE
    isSphere, radius := true, 200

    var big bool

    // Simplifiez cette structure if imbriquée
    if radius >= 200 {
        big = true
    }
```

```
    if !big {
        fmt.Println("Je ne sais pas.")
    } else if isSphere {
        fmt.Println("C'est une grosse sphère.")
    } else {
        fmt.Println("Je ne sais pas.")
    }
}
```

Résultat attendu

```
C'est une grosse sphère.
```

Exercice 3

1. Récupérez les arguments depuis la ligne de commande.
2. Utilisez une fonction ou une méthode de Go pour déterminer le nombre d'arguments.
3. Affichez les résultats attendus ci-dessous en fonction du nombre d'arguments.

Indice : Pour utiliser les arguments en Go, consultez les documentations :

- https://gobyexample.com/command-line-arguments
- https://yourbasic.org/golang/command-line-arguments/

```
package main

import (
    "fmt"
    "os"
)

func main() {
    // Votre code ici
}
```

Résultat attendu

```
go run main.go
Donnez-moi des arguments

go run main.go hello
Il y en a un : "hello"

go run main.go hi there
Il y en a deux : "hi there"

go run main.go Je veux devenir un gopher
Il y a 5 arguments
```

Exercice 4

HISTOIRE : Vous êtes curieux au sujet des échelles de Richter. Quand les journalistes disent : "Il y a eu un tremblement de terre de magnitude 5.5",

Vous voulez savoir ce que cela signifie.

Donc, vous avez décidé d'écrire un programme pour le faire à votre place.

1. Récupérez la magnitude du tremblement de terre depuis la ligne de commande.
2. Affichez sa description correspondante.

Indice : Pour utiliser les arguments en Go, consultez les documentations :

- https://gobyexample.com/command-line-arguments
- https://yourbasic.org/golang/command-line-arguments/

```go
package main

import (
    "fmt"
    "os"
)

func main() {
    // Votre code ici
}
```

Résultat attendu

MAGNITUDE	DESCRIPTION
Moins de 2.0	micro
2.0 et moins de 3.0	très mineur
3.0 et moins de 4.0	mineur
4.0 et moins de 5.0	léger
5.0 et moins de 6.0	modéré
6.0 et moins de 7.0	fort
7.0 et moins de 8.0	majeur
8.0 et moins de 10.0	énorme
10.0 ou plus	massif

Exercice 5

Refactorez le code suivant pour utilser une instruction `switch` à la place de `if` / `else`.

"Impression du nombre de jours dans un mois donné."

Indice : Pour utiliser les arguments en Go, consultez les documentations :

- https://gobyexample.com/command-line-arguments
- https://yourbasic.org/golang/command-line-arguments/

```go
package main

import (
    "fmt"
    "os"
```

```
    "strings"
    "time"
)

func main() {
    if len(os.Args) != 2 {
        fmt.Println("Donnez-moi un nom de mois")
        return
    }

    year := time.Now().Year()
    leap := year%4 == 0 && (year%100 != 0 || year%400 == 0)

    days, month := 28, os.Args[1]

    if m := strings.ToLower(month); m == "avril" ||
        m == "juin" ||
        m == "septembre" ||
        m == "novembre" {
        days = 30
    } else if m == "janvier" ||
        m == "mars" ||
        m == "mai" ||
        m == "juillet" ||
        m == "aout" ||
        m == "octobre" ||
        m == "decembre" {
        days = 31
    } else if m == "fevrier" {
        if leap {
            days = 29
        }
    } else {
        fmt.Printf("%q n'est pas un mois valide.\n", month)
        return
    }

    fmt.Printf("%q a %d jours.\n", month, days)
}
```

Exercice 6

1. À l'aide d'une boucle, additionnez les nombres de 1 à 10.
2. Affichez la somme.

```
package main

import "fmt"

func main() {
    // Votre code ici
}
```

Exercice 7

1. Étendez l'exercice "Somme jusqu'à 10".
2. Affichez les chiffres additionnés.
3. Additionnez uniquement les nombres pairs.
    - Ignorez les nombres impairs en utilisant l'instruction `continue`.

```
package main

import "fmt"

func main() {
    // Votre code ici
}
```

Résultat attendu

```
2 + 4 + 6 + 8 + 10 = 30
```

Exercice 8

- Obtenez la taille du tableau à partir de la ligne de commande.
    - Passer un nombre négatif, 0 ou invalide devrait afficher "Taille incorrecte"
    - Passer aucun argument devrait afficher "Donnez-moi la taille du tableau"
    - Passer 5 devrait créer un tableau de 5x5
    - Passer 10 pour un tableau de 10x10

Indice : Pour utiliser les arguments en Go, consultez les documentations :

- https://gobyexample.com/command-line-arguments
- https://yourbasic.org/golang/command-line-arguments/

```
package main

import (
    "fmt"
    "os"
)

func main() {
    // Votre code ici
}
```

Résultat attendu

```
go run main.go
Donnez-moi la taille du tableau

go run main.go -5
Taille incorrecte

go run main.go ABC
Taille incorrecte

go run main.go 1
X 0 1
0 0 0
```

```
1 0 1

go run main.go 2
X 0 1 2
0 0 0 0
1 0 1 2
2 0 2 4

go run main.go 3
X 0 1 2 3
0 0 0 0 0
1 0 1 2 3
2 0 2 4 6
3 0 3 6 9
```