



**Assignment 1**  
**CSI2120 Programming Paradigm**  
**Winter 2022**  
**Due March 14 11:30 pm via BrightSpace**

**8%**

Caesar Cipher

Le code César est l'un des plus anciens codes cryptographiques dans lequel chacune des lettres d'un message est décalée d'un certain nombre de positions. Les lettres du message original sont ainsi remplacées par une lettre un peu plus loin dans l'ordre alphabétique. Si la nouvelle position se retrouve au-delà de la lettre Z, alors on revient au début de l'alphabet. Par exemple, si le déplacement en position est de 5 alors toutes les occurrences de la lettre A deviendront F, les B deviendront des G, alors que les lettres Z seront décalées à E.

*The Caesar Cipher is an ancient cryptographic scheme where each letter of a message is shifted by a fixed amount. The letters of the original message are shifted to a later letter in the alphabet. If the shift amount causes a letter to be shifted beyond the letter Z, then the letter loops back to the beginning of the alphabet. For example, if the shift amount is 5, then all occurrences of the letter A would be shifted to F, all occurrences of the letter B would be shifted to G, ..., all occurrences of the letter Z would be shifted to E.*

Nous vous demandons de créer la fonction `caesar-cipher` qui accepte une chaîne de caractères (le message) et un nombre (le décalage) entre 1 et 25 inclusivement. Le message sera alors codé en appliquant le décalage alphabétique spécifié. Aussi, afin d'éviter de laisser des traces qui pourrait faciliter le décodage, tous les caractères non alphabétiques sont éliminés du message et toutes les lettres sont transformées en majuscules.

*We ask you to write a function `caesar-cipher` that consumes a string (message) and a natural number (shift) that is less than 26. The function will consume the message that will be encoded according to the shift amount given. Also, to avoid cues about the original message, all non-alphabetic characters are removed from the message, and all letters in the coded message appear as uppercase letters.*

Les fonctions Go suivantes pourraient vous être utiles :  
*The following Go functions can be useful:*

- La fonction `unicode.IsLetter(c)` vérifie si un caractère `c` est une lettre de l'alphabet. Le type Go pour les caractères unicode s'appelle `rune`, i.e. `var c rune`.

---

*The `unicode.IsLetter(c)` function checks if character `c` is a letter. The Go type for the unicode character is `rune`, i.e. `var c rune`.*

- La fonction `strings.ToUpper(s)` retourne la version majuscule d'une chaîne de caractères.

*The `strings.ToUpper(s)` function returns the uppercase version of string `s`.*

- Le programme suivant démontre comment une chaîne de caractères peut être convertie en un slice de caractères et réciproquement :

*The following program demonstrates how to convert a string into a slice of runes (characters) and reciprocally:*

```
// this program removes all occurrences of character '2' in a string
func main() {

    var s string = "Hello CSI2120" // string
    var r []rune // slice of unicode chars

    // from string to slice of unicode
    for _, c := range s {

        if c != '2' {
            r = append(r, c) // add character to slice of unicode
        }
    }

    // test
    fmt.Printf("%c : %t \n", r[7], unicode.IsLetter(r[7]))
    fmt.Printf("%c : %t \n", r[9], unicode.IsLetter(r[9]))

    // from slice of unicode to string
    var buffer bytes.Buffer
    for _, c := range r {
        buffer.WriteRune(c)
    }
    newString := buffer.String()

    fmt.Println(newString)
}
```

### **Question 1: [2 pts]**

Définir la fonction `CaesarCipher(m string, shift int) string` qui accepte un message et retourne le message encrypté. [6]

Write the `CaesarCipher(m string, shift int) string` function that accepts a message and returns the encrypted message. [6]

```
fmt.Println(CaesarCipher("I love CS!", 5))
NQTAJHX
```

**Question 2: [3 pts]**

Définir un programme Go traitant une liste de messages en utilisant une fonction concurrente. La fonction `main` de ce programme passe la liste des messages à une fonction `go` qui encrypte chacun des messages et envoie ces messages codés à un channel. La fonction `main` affiche simplement les messages encryptés dans l'ordre où ils sont reçus. [8]

*Write a Go program that processes a list of messages using a concurrent function. A main function passes the list of messages to a go function that encrypts each message and send each resulting encrypted message to a channel. The main function simply prints the encrypted messages as they are received, in any order. [8]*

```
func main() {

    // List of messages
    messages:= []string{"Csi2520", "Csi2120", "3 Paradigms",
                        "Go is 1st", "Prolog is 2nd", "Scheme is 3rd",
                        "uottawa.ca", "csi/elg/ceg/seg", "800 King Edward"}

    // Create channels???

    // call go funtion
    go CaesarCipherList(messages[:],2, _____) // send channels???

    // print results ???
    // add synchronisation ???

}
```

Le résultat produit devrait être comme suit :

*The result will look as follows:*

```
EUK
EUK
RCTCFKIOU
IQKUUV
RTQNQIKUPF
UEJGOGKUTF
WQVVCYCEC
EUKGNIEGIUGI
MKPIGFYCTF
```

**Question 3: [3 pts]**

Afin d'accélérer le traitement, nous voulons maintenant subdiviser la liste originale de messages en 3. Réécrire la fonction `main` en b) de façon à ce que 3 fonctions `go` concurrentes soient appelées, chacune traitant un tiers des données. Votre programme devrait fonctionner avec une liste de n'importe quelle dimension sans avoir à modifier votre code (sauf pour ce qui est de l'initialisation du tableau de messages). [6]

---

*To accelerate the processing, we would like to split the original list of messages into 3. Rewrite the main function in b) such that 3 concurrent go function are created. Your program should work for list of any size without having to modify the code (except the array initialization). [6]*

```
// call go funtions
go CaesarCipherList(_____) // process first 1/3 of messages???
go CaesarCipherList(_____) // process second 1/3 of messages???
go CaesarCipherList(_____) // process last 1/3 of messages???
```