IFT 1016 – Programmation I

Devoir 1

- À faire en groupe de **deux** étudiants.
- Remise : Le **19 novembre** 2019 à **23:59** au plus tard
- Il y aura une pénalité de **10%** par jour de retard.

1. Objectifs du TP1

Dans ce premier projet vous aurez l'occasion de pratiquer les concepts suivants:

• Les boucles

• Les fonctions et la décomposition fonctionnelle

2. Énoncé

Vous devez concevoir et coder deux fonctions nommées "encoder" et "decoder" qui font l'encodage et le décodage d'un texte en utilisant le cryptage de César.

Cryptage de César

Lors de ses batailles, l'empereur romain Jules César encryptait les messages qu'il envoyait à ses généraux. Sa méthode de codage consistait à décaler les lettres de 3 rangs, vers la droite, dans l'alphabet. Cette méthode de cryptage est appelée chiffrement de César. Le nombre de rangs de décalage des lettres est appelé la clef. Jules César employait donc la clef égale à 3 :

JULES CESAR $\xrightarrow{CLEF=3}$MXOHV FHVDU

mod – un opérateur modulo (%)
Le rang d'une lettre est un des entiers compris entre 0 et 25
A correspond au rang 0 mod 26
…
Z correspond au rang 25 mod 26

Crypter avec César: consiste à choisir un nombre appelé clef pour transformer une lettre d'alphabet en une autre par le jeu des transformations suivantes :

Alphabet	→	[0;25]	César (clef)	[0;25]	→	Alphabet
Lettre	→	Nombre (rang)	→	Nombre (rang)	→	Lettre
M	→	n	(n+clef) mod 26	n_c	→	M_c

Décoder avec César : utiliser la clé opposée: n_d = (n-clef) mod 26.

On peut s'arranger pour que le résultat soit toujours représenté par un entier de 0 à 25 : si n+clef (respectivement n-clef) n'est pas dans l'intervalle [0,25], il suffit de soustraire (respectivement ajouter) 26.

Vous devez faire une décomposition en sous-fonctions pour que votre code soit facile à comprendre et qu'il évite la duplication de code similaire. Vous devez avoir comme minimum 6 fonctions dans votre programme:

- `encoder`
- `decoder`
- `cesar`
- `lettre`
- `rotation`
- `tests`

Vous pouvez définir des fonctions auxiliaires au besoin.

La fonction `rotation` prend deux paramètres, un entier positif de 0 à 25, le rang de la lettre à encoder ou décoder et un entier représentant une clef d'encodage/décodage. La fonction fait la rotation. La valeur de retour de la fonction `rotation` est un nombre représentant le rang de la lettre après la rotation (par exemple, l'appel `rotation(9,3)` doit retourner `12`.

La fonction `lettre` s'occupe de l'encodage/décodage d'une seule lettre. Cette fonction doit prendre 2 arguments : un texte - lettre, un entier – clef. Pour effectuer sa tâche, la fonction `lettre` doit appeler la fonction `rotation`. La valeur à retourner est une lettre (texte) encodée/décodée.

`lettre("J",3)` doit retourner "M"

La fonction `cesar` effectue l'analyse du texte passé comme premier paramètre et envoie les symboles nécessitant les opérations d'encryptage/décryptage à la fonction `lettre`. Voici les règles de cryptage :

Seulement les lettres d'alphabet latin (26 symboles) doivent être encodées. Pour l'encodage des lettres minuscules il faudra utiliser une clé P (deuxième paramètre) et pour les majuscules - une autre clef N (troisième paramètre). Tous les autres symboles doivent être laissés sans encodage (en claire).

`cesar("JULES CESAr!!! ",5,3)` → `"MXOHV FHVDw!!! "`

Les deux fonctions `encoder` et `decoder` utilisent les trois arguments chacune, texte et deux clefs et utilisent la fonction `cesar` pour effectuer les opérations respectives.

Finalement, la fonction `tests` doit faire des tests unitaires bien choisis des fonctions `encoder`, `decoder`, `cesar`, `lettre` et `rotation`. Pour chaque fonction il faut fournir comme minimum 3 tests unitaires couvrant les cas spéciaux et le cas général :

3 tests * 5 fonctions obligatoires (`encoder`, `decoder`, `cesar`, `lettre`, `rotation`) = 15 tests minimum.

En plus de faire correctement les opérations demandées, votre code doit être correctement indenté, il doit contenir des commentaires explicatifs et des identificateurs significatifs. Il est à noter que votre code ne fait que définir des fonctions et exécuter les tests unitaires. Votre code ne doit pas faire d'entrée-sortie (pas d'appel à `print`, `alert` ou `prompt` dans la version de remise). Il faut s'imaginer que votre code sera utilisé par un autre programmeur dans un logiciel plus gros (par exemple un logiciel de traitement de texte qui doit numéroter les pages en numérotation romaine).

Dans codeBoot, sauvez votre code sous le nom de fichier "cryptage.js" (il faut double cliquer sur le nom par défaut, puis entrer le nouveau nom, puis taper la touche RETURN).

Évaluation

Ce travail compte pour 15% dans la note finale du cours. Vous devez faire le travail par groupes de 2 personnes. Indiquez vos noms clairement dans les commentaires au début de votre code.

Vous devez seulement remettre votre fichier cryptage.js

La remise doit se faire sur le site Studium du cours.

Voici les critères d'évaluation du travail :

- l'exactitude (respect de la spécification)
- l'élégance et la lisibilité du code
- la présence de commentaires explicatifs lorsque nécessaire
- le choix des identificateurs
- la décomposition fonctionnelle et le choix de tests unitaires pertinents

Indications :

- La performance de votre code doit être raisonnable.