

Devoir Surveillé - 2 h - n°1

Exercice 1 - *cetre étranger* - 2021 - sujet 1 - POO

15 points

Notion abordée : programmation objet.

Cryptage selon le « Code de César »

Dans cet exercice, on étudie une méthode de chiffrement de chaînes de caractères alphabétiques. Pour des raisons historiques, cette méthode de chiffrement est appelée "*code de César*". On considère que les messages ne contiennent que les lettres capitales de l'alphabet "ABCDEFGHIJKLMNOPQRSTUVWXYZ" et la méthode de chiffrement utilise un nombre entier fixé appelé la clé de chiffrement.

1. Soit la classe CodeCesar définie ci-dessous :

```
class CodeCesar:

    def __init__(self, cle):
        self.cle = cle
        self.alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

    def decale(self, lettre):
        num1 = self.alphabet.find(lettre)
        num2 = num1+self.cle
        if num2 >= 26:
            num2 = num2-26
        if num2 < 0:
            num2 = num2+26
        nouvelle_lettre = self.alphabet[num2]
        return nouvelle_lettre
```

On rappelle que la méthode `str.find(lettre)` renvoie l'indice (index) de la lettre dans la chaîne de caractères `str`

Représenter le résultat d'exécution du code Python suivant :

```
code1 = CodeCesar(3)
print(code1.decale('A'))
print(code1.decale('X'))
```

2. La méthode de chiffrement du « code César » consiste à décaler les lettres du message dans l'alphabet d'un nombre de rangs fixé par la clé. Par exemple, avec la clé 3, toutes les lettres sont décalées de 3 rangs vers la droite : le A devient le D, le B devient le E, etc.

Ajouter une méthode `cryptage(self, texte)` dans la classe `CodeCesar` définie à la question précédente, qui reçoit en paramètre une chaîne de caractères (le message à crypter) et qui retourne une chaîne de caractères (le message crypté).

Cette méthode `cryptage(self, texte)` doit crypter la chaîne `texte` avec la clé de l'objet de la classe `CodeCesar` qui a été instancié.

Exemple :

```
>>> code1 = CodeCesar(3)
>>> code1.cryptage("NSI")
'QVL'
```

3. Ecrire un programme qui :

- demande de saisir la clé de chiffrement
- crée un objet de classe `CodeCesar`
- demande de saisir le texte à chiffrer
- affiche le texte chiffré en appelant la méthode `cryptage`

4. On ajoute la méthode `transforme(texte)` à la classe `CodeCesar` :

```
def transforme(self, texte):
    self.cle = -self.cle
    message = self.cryptage(texte)
    self.cle = -self.cle
    return message
```

On exécute la ligne suivante : `print(CodeCesar(10).transforme("PSX"))`

Que va-t-il s'afficher ? Expliquer votre réponse.

Exercice 2 - épreuve écrite 2021 - sujet 5 : algorithme de tri par insertion

5 points

La fonction `tri_insertion` suivante prend en argument une liste `L` et trie cette liste en utilisant la méthode du tri par insertion. Compléter cette fonction pour qu'elle réponde à la spécification demandée.

```
1 def tri_insertion(L):
2     n = len(L)
3
4     # cas du tableau vide
5     if ...:
6         return L
7
8     for j in range(1,n):
9         e = L[j]
10        i = j
11
12        # A l'étape j, le sous-tableau L[0,j-1] est trié
13        # et on insère L[j] dans ce sous-tableau en déterminant
14        # le plus petit i tel que 0 <= i <= j et L[i-1] > L[j].
15        while i > 0 and L[i-1] > ...:
16            i = ...
17
18        # si i != j, on décale le sous tableau L[i,j-1] d'un cran
19        # vers la droite et on place L[j] en position i
20        if i != j:
21            for k in range(j,i,...):
22                L[k] = L[...]
23            L[i] = ...
24    return L
```

Exemples :

```
>>> tri_insertion([2,5,-1,7,0,28])
[-1, 0, 2, 5, 7, 28]
>>> tri_insertion([10,9,8,7,6,5,4,3,2,1,0])
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```