

# Travail pratique 1 : Chiffrement

Principes de Programmation - Octobre 2024 (source : KUL)

Pour envoyer des messages en toute sécurité sur internet, ceux-ci peuvent être chiffrés ou "encryptés". Dans ce travail pratique, tu vas écrire un programme capable de chiffrer un texte.

## 1 Chiffrement Caesar

Tu vas implémenter une variante de la technique de chiffrement appelée Caesar Cipher. Dans cette technique, chaque lettre est décalée d'un certain nombre de positions dans l'alphabet ("shift"). Ce décalage peut se faire vers l'avant ou vers l'arrière.

Voici un exemple d'un **chiffrement Caesar** avec un décalage de 3. Chaque caractère est décalé de 3 positions vers la droite. Le mot "jour" serait ainsi chiffré en "mxur". Pour déchiffrer le mot chiffré, tu peux appliquer la même technique avec un décalage de -3, où le signe moins indique un changement de direction.

### Exemple 1 de chiffrement de caractères avec un décalage de 3

Lettres entrées

a b c d e f g h

Lettres chiffrées

d e f g h i j k

## 2 Variante du Caesar Cipher

### 2.1 Chiffrement

Dans ce travail pratique, tu vas implémenter une variante du Caesar Cipher où chaque lettre successive est chiffrée avec un **décalage augmenté d'une unité**. Le premier caractère est décalé du nombre choisi,  $n$ , et le suivant est décalé de  $n + 1$  positions, etc.

Ton programme recevra donc, en plus du texte à chiffrer, la valeur du décalage initial  $n$  en entrée. Le texte à chiffrer se compose uniquement de lettres minuscules (a-z) et d'espaces. Les espaces ne sont pas chiffrés, mais leur position doit être prise en compte dans le calcul du chiffrement des lettres suivantes. Tu peux supposer que le décalage initial sera toujours un nombre naturel (supérieur ou égal à 0).

Ainsi, "abc" avec un décalage de 3 sera chiffré en "dfh".

## Exemple 2 d'exécution du chiffrement (C) de la variante Caesar Cipher

Lettres entrées

a b c

décalage

+3 +4 +5

Lettres chiffrées

d e f

## 2.2 Décryptage

Le décryptage du texte chiffré se fait de manière similaire, mais dans ce cas, les lettres doivent être décalées d'une unité supplémentaire dans la direction opposée afin d'obtenir le texte original. Ainsi, le texte "dfh", chiffré avec un décalage de 3, peut être décrypté en "abc", comme résumé dans le Tableau 3.

## Exemple 3 d'exécution du décryptage de la variante Caesar Cipher

Lettres entrées

d e f

décalage

-3 -4 -5

Lettres chiffrées

a b c

## 2.3 Déroulement général du programme

Tu écriras ton programme dans une fonction **cryptage**

Pour que cette fonction puisse prendre en charge à la fois le chiffrement et le décryptage, il y aura également un paramètre indiquant quelle variante doit être exécutée : c (chiffrement) ou d (décryptage).

Ta fonction recevra toujours 3 entrées :

- 1) une chaîne de caractères (le texte à chiffrer ou déchiffrer),
- 2) "e" ou "d" pour indiquer si le programme doit chiffrer ou déchiffrer,
- 3) un entier représentant la valeur du décalage initial.

Elle retourne la chaîne cryptée.

#### Exemple 4 Déroulement général du programme.

Entrées (texte à crypter / chiffrement ou décryptage / décalage initial)

```
abc  
c  
3
```

Sortie

```
dfh
```

### 2.4 Test

Tu écriras 2 fonctions tests :

- **test\_input()** afin de tester la fonction avec des données saisies au clavier
- **test\_auto()** afin de tester la fonction dans les données ci-dessous.

```
assert( test_auto(dfh, d, 3), abc)  
assert( test_auto(beginnelsen van programmeren, c, 11), mqtwcivdxh rxl psqjvfstmaoy)  
assert( test_auto(randomstring, c, 0), rbpqsryazrxr)  
assert( test_auto(python is leuk, c, 6), vfbqyy vg bvmd)  
assert( test_auto(gfkzia te rt rdeymab ninflz, d, 4), caesar is de allereerste keizer)
```

### 2.5 Conseils

Comment décaler les lettres ? Chaque lettre correspond à un nombre, que tu peux obtenir grâce à la fonction intégrée de Python **ord()**. Par exemple, `ord('a')` donne 97. Les lettres consécutives correspondent à des nombres consécutifs, donc `ord('b')` donne 98, etc.

Inversement, tu peux convertir un nombre en la lettre correspondante avec **chr()**. Par exemple, `chr(98)` donne 'b', `chr(97)` donne 'a', etc.

Réfléchis donc à la formule qui te permet de décaler une lettre d'un certain nombre de positions. Tu peux te limiter aux lettres minuscules (a-z).