

César Code

Bienvenue à ce coding club sur le thème de la **cryptographie** !

Aujourd'hui, nous allons explorer l'un des algorithmes de cryptographie les plus anciens et les plus simples, le *chiffrement de César*.

Nous allons découvrir comment ce chiffrement fonctionne, comment il peut être implémenté en utilisant différents langages de programmation et comment il a été utilisé dans l'histoire pour sécuriser des messages importants.

En fin de compte, nous espérons que cette session de coding club vous donnera une meilleure compréhension de la cryptographie et de la manière dont les algorithmes de chiffrement sont utilisés pour protéger les données sensibles.

Introduction

2042, la Terre est attaquée par des **Extra-Terrestres**.

Marvin, notre robot super intelligent (*Approximativement 30 milliards de fois plus intelligent que n'importe quel humain*) a intercepté des messages top-secrets provenant des **Extra-Terrestres**.

Votre mission, *si toute fois vous l'accepter*, sera de **déchiffrer le message** ennemi et de **le remplacer par un message chiffré** qui les menera à la défaite.

Votre rôle dans cette guerre est primordial.

Installation

Ce coding-club est disponible en 3 langages de programmation. Merci de vous assurer d'avoir pris le bon sujet.

Si vous souhaitez faire ce coding club dans un autre langage de programmation sachez qu'il est disponible en :

- C
- C++
- Python

Si vous souhaitez travailler en Python :

- Rendez-vous dans le dossier SUJET_PYTHON

Vous trouverez un fichier Exercice_01.py et un fichier Exercice_02.py à compléter dans leur dossier respectif.

Pour essayer votre programme utilisez la commande suivante :

```
./chiffrement [FICHIER] [CLE]
./dechiffrement [FICHIER] [CLE]
```

Si vous souhaitez travailler en C :

- Rendez-vous dans le dossier SUJET_C

Vous trouverez un fichier Exercice_01.c et un fichier Exercice_02.c à compléter dans leur dossier respectif.

Pour compiler et exécuter votre code, vous pouvez utiliser la commande suivante :

```
make
```

Pour essayer votre programme utilisez la commande suivante :

```
./chiffrement [FICHIER] [CLE]
./dechiffrement [FICHIER] [CLE]
```

Si vous souhaitez travailler en C++ :

- Rendez-vous dans le dossier SUJET_CPP

Informations

Si vous rencontrez des difficultés ou que vous avez une question **n'hésitez pas à appeler un Cobra**. Ils sont là pour vous accompagner !

Une fiche d'aide est disponible à la fin de ce sujet pour vous accompagner à travers ce coding-club.

Message à déchiffrer : "Rovvy Gybvñ !"
Clé pour déchiffrer ce message : 16
Message déchiffré : "Hello World !"

Message à chiffrer : "Hello World !"
Clé pour chiffrer ce message : 23
Message chiffré : "Ebiil Tloia !"

Exercice 1 : Déchiffrement

Marvin vous a transmis les messages qu'il a intercepté ! Votre mission est de les déchiffrer pour en déterminer le contenu !

Dans cet exercice l'objectif est d'écrire la fonction **dechiffrement**.

Exercice 2 : Chiffrement

Marvin vous demande maintenant de remplacer le message intercepté par un faux message chiffré !

Dans cet exercice l'objectif est d'écrire la fonction **chiffrement**.

Conclusion

Pseudo-Code

Marvin vous a transmis le *pseudo-code* permettant d'écrire l'algorithme pour déchiffrer un message, le voici :

- Pour chaque caractère [C] dans le message:
 - Si [C] est une majuscule:
 - Calculer le rang dans l'alphabet [R] de [C] dans l'alphabet majuscule
 - Calculer le nouveau rang [R2] = ([R] - [D]) % 26
 - Afficher le caractère [C2] = ASCII('A') + [R2]
 - Si [C] est une miniscule:
 - Calculer le rang dans l'alphabet [R] de [C] dans l'alphabet majuscule
 - Calculer le nouveau rang [R2] = ([R] - [D]) % 26
 - Afficher le caractère [C2] = ASCII('a') + [R2]
 - Si [C] n'est pas une lettre:
 - Afficher le caractère [C]