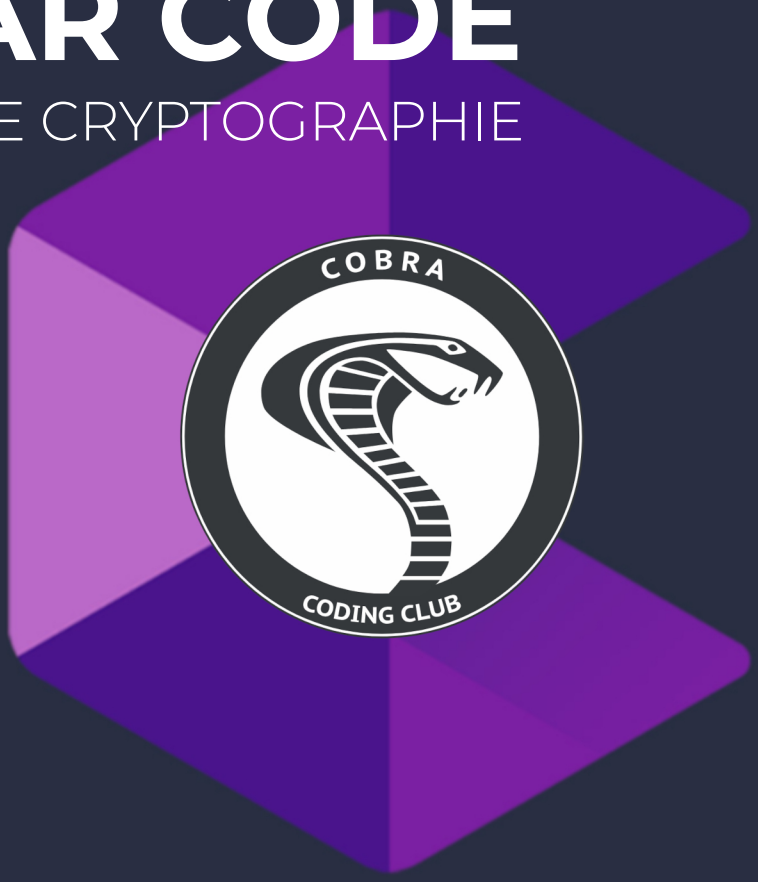
DIVERSITY by EPITECH

CESAR CODE

ATELIER DE CRYPTOGRAPHIE



CESAR 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.

Objectifs

- ✓ Comprendre les bases de la cryptographie et du chiffrement.
- ✓ Apprendre à implémenter l'algorithme de chiffrement de César en code.
- ✓ Développer des compétences en programmation (Python, C ou C++).
- ✓ Renforcer la logique algorithmique et la résolution de problèmes.

Installation

Si vous êtes sur votre machine personnelle, vous pouvez installer les outils nécessaires pour ce coding club en suivant les instructions ci-dessous.

Ordinateur personnel

Python

Pour installer Python sur votre machine, vous pouvez suivre les instructions sur le site officiel de Python : `https://www.python.org/downloads/`

C

Pour installer un compilateur C sur votre machine, vous pouvez suivre les instructions sur le site officiel de GCC : `https://gcc.gnu.org/install/`

C++

Pour installer un compilateur C++ sur votre machine, vous pouvez suivre les instructions sur le site officiel de GCC : `https://gcc.gnu.org/install/`

Visual Studio Code

Pour installer Visual Studio Code sur votre machine, vous pouvez suivre les instructions sur le site officiel de Visual Studio Code : `https://code.visualstudio.com/download`

Ordinateur Cobra

L'ensemble des outils nécessaires pour ce coding club sont déjà installés sur les ordinateurs Cobra. Vous devez avoir sur le bureau un dossier nommé **Coding Club**. Ce dossier contient les outils nécessaires pour ce coding club.

Si vous rencontrez des difficultés lors de l'installation des outils, n'hésitez pas à demander de l'aide à l'équipe Cobra.

Introduction à la cryptographie et au chiffrement de César

La cryptographie est l'art et la science de sécuriser les communications en convertissant des informations en un format illisible pour quiconque n'a pas la clé de déchiffrement appropriée. Son objectif principal est de garantir la confidentialité, l'intégrité et l'authenticité des données, ainsi que de permettre des opérations sécurisées dans un monde de plus en plus interconnecté.

Importance de la Cryptographie :

La cryptographie joue un rôle crucial dans la sécurité informatique, offrant plusieurs avantages importants :

Confidentialité : Elle permet de cacher les informations réelles, accessibles uniquement par les personnes avec la clé de déchiffrement. C'est vital pour des transactions financières en ligne ou des échanges diplomatiques.

Intégrité : La cryptographie évite les altérations non autorisées des données pendant leur transmission. Toute modification est détectée, assurant que les données restent fiables.

Authenticité : Elle vérifie l'origine des données. Les signatures numériques garantissent que les données proviennent bien de l'expéditeur prétendu et n'ont pas été changées.

Contrôle d'accès : Elle gère les accès et permissions. Les méthodes d'authentification, comme les mots de passe et les certificats numériques, sécurisent les ressources contre un accès non autorisé.

Protection des mots de passe : Les mots de passe sont stockés sous forme de hachages cryptographiques, renforçant la sécurité, même lors de fuites de données.

Sécurité des transactions : Dans les paiements en ligne, la cryptographie protège les données sensibles, comme les numéros de carte de crédit, des interceptions et abus.

Protection de la vie privée : Elle préserve la confidentialité des communications en ligne, assurant la sécurité des échanges et la vie privée dans le monde numérique.

En résumé, la cryptographie est essentielle à la sécurité informatique. Elle établit la confiance dans les transactions en ligne et les échanges numériques, garantissant la confidentialité, l'intégrité et l'authenticité des données, et maintenant ainsi la sécurité et la confiance dans notre monde numérique.

Histoire du chiffrement de César

Le chiffrement de César, inspiré par Jules César, est une technique simple où les lettres d'un message sont remplacées par d'autres lettres décalées dans l'alphabet. Pour chiffrer, on choisit un "décalage" en décidant combien de lettres on avance dans l'alphabet.

Par exemple, avec un décalage de 3, A devient D, B devient E, et ainsi de suite. Ce système était utilisé autrefois pour garder des messages secrets dans l'armée, mais il est facile à décoder car il y a seulement 26 décalages possibles (pour les 26 lettres).

Bien qu'il ne soit plus très sécurisé, le chiffrement de César est toujours enseigné pour expliquer les bases de la sécurité informatique et de la cryptographie.

Fonctionnement du chiffrement de César

Le chiffrement de César est une méthode très simple de cryptographie. Voici comment ça marche :

Choix de la clé (décalage) : Vous choisissez un nombre qui indique combien chaque lettre du message original sera décalée dans l'alphabet. Par exemple, avec un décalage de 3, A devient D, B devient E, et ainsi de suite.

Conversion du message : Prenez le message à chiffrer et appliquez le décalage à chaque lettre. Si le décalage vous fait sortir de l'alphabet, revenez au début. Par exemple, si le décalage est de 3 et que vous arrivez à Z, retournez à C.

Chiffrement : Remplacez chaque lettre du message par la lettre correspondante après le décalage. Les autres caractères restent inchangés.

Message chiffré : Le résultat est le message chiffré, composé des lettres décalées.

Exemple : Si le mot original est "HELLO" et que le décalage est de 3, chaque lettre est déplacée de 3 places, donnant le message chiffré "KHOOR".

Il est important de savoir que le chiffrement de César est vulnérable aux attaques par force brute, car il n'y a que 25 décalages possibles. Cela signifie qu'il ne peut pas protéger efficacement les informations secrètes dans le monde moderne. Malgré cela, il est enseigné pour comprendre les bases de la sécurité informatique et de la cryptographie.

Introduction

En 2042, un voile sombre de danger s'étendit sur la Terre alors qu'elle était attaquée par des forces extra-terrestres. Face à cette menace sans précédent, une lueur d'espoir brilla sous la forme de Marvin, un robot surdoué dont l'intelligence surpassait de loin celle de tous les humains combinés, environ 30 milliards de fois.

Marvin, dans sa prodigieuse clairvoyance, intercepta des messages cryptés en provenance des extra-terrestres. Ces messages, remplis de secrets et de stratégies hostiles, furent une lueur d'espoir dans l'obscurité. Mais l'espoir ne pouvait se réaliser sans l'aide cruciale de ceux qui oseraient prendre la mission en main.

C'est ici que vous entrez en scène, acceptant le défi d'une mission qui pourrait bien définir le destin de notre monde. Votre mission, aussi monumentale que délicate, est de décoder ces messages ennemis. Mais ce n'est pas tout – vous devez également forger un message chiffré, soigneusement conçu pour semer la confusion parmi les extra-terrestres et les mener vers leur propre défaite.

Votre rôle, dans cette guerre épique pour la survie de l'humanité, est d'une importance primordiale. Chaque action, chaque choix, pourrait être le déclencheur de la victoire. L'avenir de la Terre repose entre vos mains, aux côtés de Marvin et de l'équipe formée pour cette mission audacieuse.

Alors que vous vous embarquez dans cette aventure incroyable, rappelez-vous que même dans les moments les plus sombres, le courage et l'ingéniosité humaine peuvent briller d'une manière extraordinaire. Vous n'êtes pas seuls dans cette bataille. Vous êtes soutenus par Marvin, le robot super intelligent, et par le potentiel infini qui réside dans votre propre esprit.

Que cette épopée de détermination et de stratégie commence. Les étoiles peuvent sembler lointaines et hostiles, mais c'est précisément dans ces moments de défi que l'humanité trouve sa force intérieure la plus profonde. Votre mission est lancée, et avec elle, l'espoir de la Terre de préserver sa place dans le cosmos.

Lecture d'un message et décalage

Etape 1 : Lecture, stockage d'un message et affichage

Vous venez d'intercepter un message, vous devez écrire un programme informatique pour le déchiffrer. Pour cela, vous allez devoir apprendre comment chiffre un message, puis comment le déchiffrer avec un décalage connu et enfin comment le déchiffrer sans connaitre le décalage.

1. Stocker le message suivant dans une variable.

✓ **Message** : "Bonjour, je suis Marvin, votre robot super intelligent."

2. Afficher le message sur la console.

```
Terminal
Cobra> ./cesar_code
Bonjour, je suis Marvin, votre robot super intelligent.
```

3. Demander à l'utilisateur de saisir un message et le nombre pour le décalage. Stockez les informations dans des variables, puis affichez les informations.

Pour demander à l'utilisateur de saisir un message, vous pouvez utiliser la fonction `input()` en Python, `scanf()` en C et `cin` en C++.

```
message = input("Entrez le message a chiffrer : ")
decalage = int(input("Entrez le decalage : "))
```

Voici un exemple d'affichage :

```
Terminal
Cobra> ./cesar_code
Saissisez le message que vous souhaitez chiffrer :
Bonjour, je suis Marvin, votre robot super intelligent.
Saissisez le décalage que vous souhaitez utiliser :
3
Vous avez saisi le message suivant : Bonjour, je suis Marvin, votre robot super
intelligent. Ce message sera chiffré avec un décalage de 3.
```

Etape 2 : Chiffrer le message

Félicitations, vous avez réussi à stocker le message et le décalage dans des variables. Maintenant, vous allez devoir chiffrer le message en utilisant le décalage.

1. Parcourir le message lettre par lettre et afficher chaque lettre sur la console en prennant en compte que les caractères alphanumérique.

```
Terminal
Cobra> ./cesar_code “Bonjour, je suis Marvin, votre robot super intelligent.”
B
o
n
j
o
u
r
j
e
... etc
```

Pour parcourir une chaîne de caractères, vous pouvez utiliser une boucle `for` ou `while` en Python, C et C++.

Pour connaître la longueur d'une chaîne de caractères, vous pouvez utiliser la fonction `len()` en Python, `strlen()` en C et `length()` en C++.

Nous vous invitions à recoder la fonction `**isalpha()**` pour vérifier si un caractère est alphanumérique.

2. Pour chaque lettre, ajouter le décalage au code ASCII de la lettre.
   Vous devez ajouter le décalage au code ASCII de la lettre afin d'obtenir le code ASCII de la lettre chiffrée.

ASCII est un code de caractères qui associe un entier à chaque symbole. Par exemple, le code ASCII de "**A**" est **65**, le code ASCII de "**B**" est **66**, etc.
Vous pouvez utiliser la fonction `ord()` en Python, `scanf()` en C et `cin` en C++ pour obtenir le code ASCII d'un caractère.
La table ASCII est disponible sur Wikipedia ou via la commande `man ascii` sur un terminal Linux.

Exemple avec le message "**Bonjour**" et un décalage de **3** :

- ✓ **Message original**: BONJOUR
- ✓ **Décalage**: 3
- ✓ **Message chiffré**: ERQMRXU

```
Terminal
Cobra> ./cesar_code ‘‘Bonjour’’ 3
ERQMRXU
```

3. Gérez les cas des caractères non alphanumériques. Vous devez ignorer les caractères non alphanumériques et les afficher tels quels.

- ✓ **Message original**: Hello, World!
- ✓ **Décalage**: 3
- ✓ **Message chiffré**: Khoor, Zruog!

4. Gérez les cas où le code ASCII dépasse la valeur de 122 (code ASCII de la lettre **z**). Dans ce cas, vous devez revenir au début de l'alphabet.

Etape 3 : Déchiffrer le message

Bravo, vous avez réussi à chiffrer le message. Maintenant, vous allez devoir déchiffrer le message en utilisant le décalage.

1. Parcourir le message lettre par lettre et afficher chaque lettre sur la console en prennant en compte que les caractères alphanumérique.
2. Pour chaque lettre, soustraire le décalage au code ASCII de la lettre.

Exemple avec le message "**ERQMRXU**" et un décalage de **3** :

- ✓ **Message chiffré**: Khoor, Zruog!

✓ **Décalage**: 3
✓ **Message original**: Hello, World!

Etape 4 : Déchiffrer le message sans connaitre le décalage

Lorsque vous interceptez un message, vous ne connaissez pas le décalage utilisé pour le chiffrer. Vous allez devoir trouver le décalage en explorant d'autres méthodes.
Voici une liste non-exhaustive de méthodes que vous pouvez utiliser pour trouver le décalage :

✓ **Fréquence des Lettres** : Les langues ont des lettres fréquentes. En analysant la fréquence des lettres dans le message chiffré, les élèves pourraient repérer des correspondances avec les fréquences typiques de la langue. wikipedia

✓ **Mots Courants** : Les mots courants apparaissent régulièrement. En identifiant des mots fréquents comme "the", "and" ou "is", les élèves pourraient déduire le décalage à partir des lettres correspondantes.wikipedia

✓ **Dictionnaires de Mots** : Les dictionnaires peuvent aider à trouver des mots reconnus dans le message chiffré. En itérant à travers les mots du dictionnaire, les élèves pourraient repérer des correspondances.

✓ **Approche de Force Brute** : Si le décalage est petit, les élèves pourraient essayer toutes les possibilités de décalage jusqu'à ce que le message ait du sens.wikipedia

Annexes

Bases du langage python

Python est un langage de programmation de haut niveau réputé pour sa syntaxe claire et lisible. Les blocs de code sont délimités par l'indentation, ce qui signifie que l'espacement correct est essentiel. Les déclarations se terminent généralement par des sauts de ligne, mais vous pouvez également utiliser un point-virgule pour les séparer.

Variables et Types de Données :

En Python, vous pouvez créer des variables pour stocker différentes sortes de données, comme des nombres, des chaînes de caractères, des listes, des dictionnaires, etc. Les noms de variables sont sensibles à la casse et doivent commencer par une lettre ou un soulignement (_). Exemples de types de données :

```
nombre_entier = 42
nombre_decimal = 3.14
chaine = "Bonjour, Python!"
liste = [1, 2, 3, 4]
dictionnaire = {'cle': 'valeur'}
```

Opérations de Base :

Python prend en charge les opérations mathématiques de base (+, -, *, /) ainsi que les opérations logiques (and, or, not). Exemples :

```
somme = 5 + 3
difference = 10 - 2
produit = 4 * 6
quotient = 12 / 3

vrai_et_faux = True and False
vrai_ou_faux = True or False
non_vrai = not True
```

Structures de Contrôle :

Python propose des structures de contrôle telles que les boucles (for, while) et les instructions conditionnelles (if, elif, else) pour gérer le flux d'exécution du programme.

```
for i in range(5):
print(i)  # Affiche les nombres de 0 a 4

x = 0
while x < 5:
    print(x)
    x += 1  # Incremente x a chaque iteration

if nombre > 10:
    print("Nombre superieur a 10")
elif nombre == 10:
    print("Nombre egal a 10")
else:
    print("Nombre inferieur a 10")
```

Fonctions :

Les fonctions sont des blocs de code réutilisables qui effectuent une tâche spécifique. Elles peuvent prendre des paramètres en entrée et renvoyer une valeur en sortie.

```
def carre(x):
    return x * x
```

Listes et boucles :

Les listes sont des collections ordonnées d'éléments. Elles peuvent contenir des éléments de différents types, et vous pouvez ajouter ou supprimer des éléments à tout moment. Les boucles for sont souvent utilisées pour parcourir les listes.

```
nombre = [1, 2, 3, 4, 5]
for nombre in nombres:
    print(nombre)
```

Bases du langage C

Le langage C est un langage de programmation procédural qui a été développé à la fin des années 1960. Il est réputé pour son efficacité, sa portabilité et sa capacité à manipuler directement la mémoire de l'ordinateur.

Le code C est organisé en fonctions, où chaque fonction effectue une tâche spécifique. Les blocs de code sont délimités par des accolades `{}`. Les déclarations se terminent par un point-virgule `;`.

Variables et Types de Données :

En C, vous pouvez déclarer des variables pour stocker différentes sortes de données, comme des entiers, des nombres à virgule flottante, des caractères, etc. Les noms de variables sont sensibles à la casse.

```
int entier = 42;
float decimal = 3.14;
char caractere = 'A';
```

Opérations de Base :

C prend en charge les opérations mathématiques de base (+, -, *, /) ainsi que les opérations logiques (&&, ||, !). Exemples :

```
int somme = 5 + 3;
int difference = 10 - 2;
int produit = 4 * 6;
float quotient = 12.0 / 3.0;

int vrai_et_faux = 1 && 0;
int vrai_ou_faux = 1 || 0;
int non_vrai = !1;
```

Structures de Contrôle :

C propose des structures de contrôle telles que les boucles (for, while) et les instructions condi-

tionnelles (if, else) pour gérer le flux d'exécution du programme.

```
for (int i = 0; i < 5; i++) {
    printf("%d\n", i);  // Affiche les nombres de 0 a 4
}

int x = 0;
while (x < 5) {
    printf("%d\n", x);
    x++;  // Incremente x a chaque iteration
}

int nombre = 15;
if (nombre > 10) {
    printf("Nombre superieur a 10\n");
}
else if (nombre == 10) {
    printf("Nombre egal a 10\n");
}
else {
    printf("Nombre inferieur a 10\n");
}
```

Fonctions :

Les fonctions en C permettent de regrouper du code réutilisable. Une fonction peut accepter des paramètres et retourner une valeur.

```
int additon(int nb1, int nb2) {
    return nb1 + nb2;
}
```

Tableaux et boucles :

Les tableaux en C permettent de stocker plusieurs valeurs du même type. Ils sont indexés à partir de zéro.

```
int nombres[4] = {1, 2, 3, 4};
for (int i = 0; i < 4; i++) {
    printf("%d\n", nombres[i]);
}

// Sortie :
// 1
// 2
// 3
// 4
```

Bases du langage Javascript

JavaScript est un langage de programmation de haut niveau principalement utilisé pour créer des sites web interactifs et dynamiques. Il est exécuté côté client, ce qui signifie que le code est exécuté dans le navigateur web de l'utilisateur. Il est possible de l'exécuter côté serveur en utilisant Node.js.

La syntaxe JavaScript est similaire à celle de nombreux autres langages de programmation. Les déclarations se terminent par un point-virgule `;`. Les blocs de code sont délimités par des accolades `{}`. Les noms de variables sont sensibles à la casse.

Variables et Types de Données :

En JavaScript, vous pouvez déclarer des variables pour stocker différentes sortes de données, comme des nombres, des chaînes de caractères, des booléens, etc. Les noms de variables sont sensibles à la casse.

```
let entier = 42;
let decimal = 3.14;
let chaine = "Bonjour, JavaScript!";
let estVrai = true;
```

Opérations de Base :

JavaScript prend en charge les opérations mathématiques de base (+, -, *, /) ainsi que les opérations logiques (&& pour "et", || pour "ou", ! pour "non").

```
let somme = 5 + 3;
let difference = 10 - 2;
let produit = 4 * 6;
let quotient = 12 / 3;

let vraiEtFaux = true && false;
let vraiOuFaux = true || false;
let nonVrai = !true;
```

Structures de Contrôle :

JavaScript propose des structures de contrôle similaires à celles d'autres langages, comme les boucles (for, while) et les instructions conditionnelles (if, else if, else).

```
for (let i = 0; i < 5; i++) {
    console.log(i);  // Affiche les nombres de 0 a 4
}

let x = 0;
while (x < 5) {
    console.log(x);
    x++;  // Incremente x a chaque iteration
}

let nombre = 15;
if (nombre > 10) {
    console.log("Nombre superieur a 10");
}
else if (nombre === 10) {
    console.log("Nombre egal a 10");
}
else {
    console.log("Nombre inferieur a 10");
}
```

Fonctions :

Les fonctions en JavaScript permettent de regrouper du code réutilisable. Une fonction peut accepter des paramètres et retourner une valeur.

```
function ajouter(a, b) {
    return a + b;
}

let resultat = ajouter(3, 7);  // resultat vaut 10
```

Objets et Propriétés :

JavaScript est basé sur un modèle objet. Vous pouvez créer des objets avec des propriétés et des méthodes.

```
let personne = {
```

```
    nom: "Jean",
    age: 30,
    sePresenter: function() {
        console.log("Bonjour, je m'appelle " + this.nom + " et j'ai " + this.age + " ans
            .");
    }
};

personne.sePresenter();  // Affiche "Bonjour, je m'appelle Jean et j'ai 30 ans."
```

DIVERSITY
by EPITECH