La machine d'énigme

420-C21-IN Programmation II

Évaluation **sommative,** remise le **12 mars 2023**

Directive : Vous devez remettre votre projet sur le dépôt Moodle **Projet de mi-session** du cours

Objectif du laboratoire

Durant la Seconde Guerre, le Troisième Reich utilisait la machine enigma. Elle permettait de crypter un message à son point de départ et de décrypter le message à son point d'arrivée. Ainsi, lorsque les messages étaient interceptés, il n'était pas possible de connaître le message original.

Dans ce projet, vous devez développer qui simule une version simplifiée de la machine enigma. Votre programme reposera sur le code de César que nous avons vu en classe. Pour y parvenir, vous devrez donc développer de zéro. Pour y parvenir, vous devez :

- Maîtriser les structures
- Utiliser les fonctions
- Gérer les passages de paramètre

Fonctionnement de la machine

Contrairement à l'exemple vu en classe, notre machine utilisera quatre clés de chiffrement. Comme pour le code de César, les clés permettent de décaler les lettres du message. Toutefois, les clés seront appliquées successivement sur le message précédent. Ainsi, la première clé s'applique sur le message en clair. Par la suite, la deuxième clé s'applique sur le message décaler par la première clé et ainsi de suite. En somme, il s'agit d'appliquer le code de César quatre fois.

Exemple de chiffrement de la machine énigme
La quatrième clé a été ignoré pour simplifier le schéma

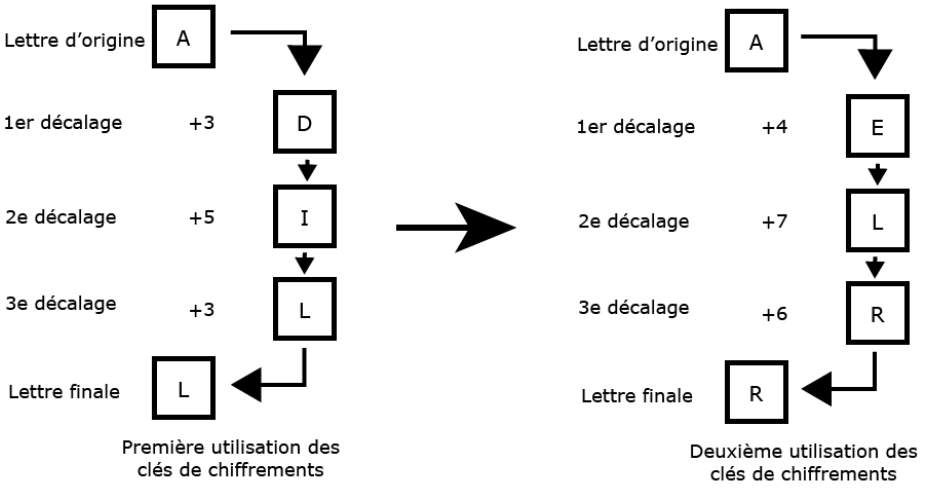

Figure 1 Illustration du fonctionnement de la machine

Néanmoins, cette version du code de César est plus sécuritaire que le code original, mais il est toutefois facile à craquer. Nous devons donc appliquer un second effet. Comme pour la machine enigma, nos clés subiront un décalage à chaque utilisation. L'idée est fort simple. À chaque utilisation de la clé, nous modifions la clé pour qu'elle avance dans le décalage. C'est dans la phase de préparation que vous dériverez l'avancement des unités de chiffrement.

Dans l'exemple de la Figure 1, les décalages des quatre clés étaient les suivants :

- La première clé avance d'une unité.
- La deuxième clé avance de deux unités.
- La troisième clé avance de trois unités.

Bien évidemment, pour compléter ce projet, vous devez appliquer une quatrième clé. De plus, votre logiciel doit crypter et décrypter les messages.

Spécifications techniques

Variables globales

Vous ne devez pas utiliser les variables globales. Leur utilisation entraînera un échec automatique.

Fonctions

Vous devez créer au moins quatre fonctions sans compter la fonction **main**. Si vous avez moins de quatre fonctions, votre note ne pourra pas dépasser 40 %.

Une fonction, une action

Vous devez utiliser la programmation de fonction selon le principal ***Une fonction, une action***. Votre fonction **main** servira uniquement à appeler les autres fonctions. Cette fonction sert aussi à lire et afficher sur la console. Si vous ne respectez pas ce principe, votre note ne pourra pas dépasser 60 %.

Structure

Vous devez créer au moins deux structures ***pertinentes***. De plus, vous ne pouvez pas utiliser de type de base autre que dans les structures. Si vous utilisez une variable ou plus avec un type de base, autre que dans une structure, votre note ne pourra pas dépasser 40 %.

Interface

Vous devez créer une interface pertinente. Elle ne doit pas se résumer à afficher simplement les éléments interactifs.

Paramètres

Vous devez utiliser, avec justesse, le passage par référence dans vos fonctions. Vous serez noté sur la pertinence de son utilisation. De plus, c'est à vous de choisir les bons types. N'hésitez pas à utiliser des paramètres constants.

Exemple

Pour vous aider dans la vérification, voici un exemple de cryptage :

Phrase originale : Star Trek, Star Wars, c'est aussi bon l'un que l'autre... sauf que Star Trek est meilleur.

Phrase cryptée : Sucu Yxls, Dfnf Mrjl, y'crt cxwxo jxx x'ic hmx g'xssrf... yhco bgr Hjrj Nmah dsu pinrsmdb.

	Clé 1	Clé 2	Clé 3	Clé 4
Clés utilisées[1]	7	15	24	6
Décalage utilisé	1	2	3	21

[1] Il s'agit de la valeur de départ de la clé.

Remise

Vous devez remettre deux fichiers sur le dépôt **Moodle** du cours avant la date limite indiquée dans le dépôt :

- Votre code avec le nom *C21-PMS-Nom-Prenom.cpp*
  - Tout votre programme doit être contenu dans un seul fichier.
- L'exécutable de votre programme *C21-PMS-Nom-Prenom.exe*.

La remise doit avoir lieu au plus tard le **12 mars 2023 à 23h59**. Les travaux sont acceptés jusqu'au **22 mars 2023 à 15h30**.

Afin d'être équitables envers toutes les personnes apprenantes du cours, les travaux remis en retard auront une pénalité de ***10 % par jour de retard***.

En conformité avec la politique institutionnelle d'évaluation des apprentissages ainsi que les règles d'encadrement du département, chaque erreur de français entraîne une pénalité de ***0,6 point*** pour une pénalité maximale de ***10 % (6 points)***. Cette règle ne s'applique pas au contenu de programmation (excluant les textes affichés).

Les pénalités de retard et de français sont calculées sur la note finale du travail.

Critères d'évaluation

Ce projet a une pondération de 25 % de la note finale. La note du travail sera évaluée à partir des critères suivants qui suivent. De plus, vous pourriez être soumis à différentes pénalités. Le total des points sera de 60 points selon la réparation suivante :

- 35 points pour l'algorithme et la programmation en général.
- 14 points pour les fonctions (déclaration, définition et paramètre)
- 6 points pour les structures (définition et pertinence)
- 5 points pour l'interface

De plus, vous pourriez avoir des pénalités en cas de non-respect des critères :

- Moins de quatre fonctions (max 40 % - 24 points)
- Non-respect du principe ***Une fonction, une action*** (max 60 % - 36 points)
- Moins de deux structures pertinentes (max 40 % - 24 points)

Plan de réalisation (ébauche)

Réalisez le projet étape par étape. Ceci vous permettra de valider vos sources et vous idées.

1. Planifier les fonctions et les structures
2. Implanter l'algorithme de César
3. Implanter le décalage
4. Implanter les fonctions
5. Implanter les structures
6. Etc.