

I2010 : langage C (13)

Modularisation

Le but de cet exercice est de construire 4 applications en C qui « chiffrent » ou « déchiffrent » une chaîne de caractères.

Une application qui chiffre une chaîne de caractères transforme cette chaîne en une chaîne incompréhensible. Une application de déchiffrement est nécessaire pour transformer la chaîne incompréhensible de manière à retrouver la chaîne compréhensible.

Dans cet exercice, nous nous intéressons à deux méthodes rudimentaires de chiffrement :

1. ROT13 (cf. <https://en.wikipedia.org/wiki/ROT13>)
2. Carré de Polybe (cf. https://fr.wikipedia.org/wiki/Carr%C3%A9_de_Polybe) utilisant le carré ci-dessous :

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|----|---|---|---|---|---|---|---|
| 1 | A | B | C | D | E | F | G | H |
| 2 | I | J | K | L | M | N | O | P |
| 3 | Q | R | S | T | U | V | W | X |
| 4 | Y | Z | 0 | 1 | 2 | 3 | 4 | 5 |
| 5 | 6 | 7 | 8 | 9 | | ! | " | # |
| 6 | \$ | % | & | ' | (|) | * | + |
| 7 | , | - | . | / | : | ; | < | = |
| 8 | > | ? | @ | [| \ |] | ^ | _ |

Fichier d'en-tête

Dans un premier temps, nous vous demandons de définir un fichier d'entête « `crypt.h` » qui **définit** et **spécifie** les deux fonctions :

1. `crypt` de type `char*` qui prend une chaîne de caractères en paramètre et renvoie la chaîne cryptée.
2. `decrypt`, de type `char*` qui prend une chaîne de caractères en paramètre et renvoie la chaîne décryptée.

¹ https://fr.wikipedia.org/wiki/Carr%C3%A9_de_Polybe

Fichier source

Dans un second temps, nous vous demandons de construire deux implémentations du fichier « `crypt.h` ». Une implémentation « `rot13.c` » relative à la méthode ROT13 et une implémentation « `polybe.c` » relative à la méthode du carré de Polybe.

Remarquez qu'avec la méthode ROT13, seules les lettres de l'alphabet sont cryptées, tout autre caractère reste inchangé. Quant à la méthode Polybé, nous supposons que les phrases à crypter ne sont composées que des caractères présents dans le carré ci-dessus.

Makefile

Dans un troisième temps, nous vous demandons de définir un fichier « `Makefile` » qui permet de générer les deux fichiers objets « `rot13.o` » et « `polybe.o` ».

Applications

Dans un quatrième temps, nous vous demandons de construire les deux applications suivantes :

1. L'application « `encrypt.c` » qui prend en argument une chaîne de caractères et qui affiche sur `stdout` la version chiffrée de cette chaîne.
2. L'application « `decrypt.c` » qui prend en argument une chaîne de caractères chiffrée et qui affiche sur `stdout` la version non-chiffrée de cette chaîne.

Makefile (suite)

Dans un cinquième temps, nous vous demandons de modifier votre fichier « `Makefile` » pour qu'il permette de générer les 4 applications suivantes :

1. L'application « `erot13` » qui prend en argument une chaîne de caractères et qui affiche sur `stdout` la version chiffrée à l'aide de la méthode ROT13 de cette chaîne.
2. L'application « `drot13` » qui prend en argument une chaîne de caractères chiffrée à l'aide de la méthode ROT13 et qui affiche sur `stdout` la version non-chiffrée de cette chaîne.
3. L'application « `epolybe` » qui prend en argument une chaîne de caractères et qui affiche sur `stdout` la version chiffrée à l'aide de la méthode du carré de Polybe de cette chaîne.
4. L'application « `dpolybe` » qui prend en argument une chaîne de caractères chiffrée à l'aide de la méthode du carré de Polybe et qui affiche sur `stdout` la version non-chiffrée de cette chaîne.