

CC1 – Programmation niveau 1

1 – Exercice I : Deviner le bon nombre

L'objectif de cet exercice est de vous faire développer un programme qui devinera un nombre en plusieurs étapes successives. Disons par exemple que le "nombre à deviner" soit égal à 10, et qu'un utilisateur essaie de le deviner. Il y a trois cas de figure possible :

- Le nombre entré par l'utilisateur est **supérieur** au nombre à deviner, ce qui entraînera une réponse du type "le nombre à deviner est inférieur au nombre entré. Réessayez.",
- Le nombre entré par l'utilisateur est **inférieur** au nombre à deviner, ce qui entraînera une réponse du type "le nombre à deviner est supérieur au nombre entré. Réessayez.",
- Le nombre entré par l'utilisateur est **égal** au nombre à deviner, ce qui causera la fin du jeu.

Dans les deux premiers cas de figure, le programme demandera à l'utilisateur de rentrer un autre nombre, et ce jusqu'à ce que l'utilisateur trouve le nombre à deviner.

2 – Exercice II : Le chiffrement de César

L'objectif de cet exercice est de vous faire chiffrer et déchiffrer des messages en utilisant le chiffrement de César. Cette méthode viendrait de Julius César, qui l'aurait utilisée pour chiffrer ses communications privées.

Le chiffrement de César est un algorithme cryptographique par substitution où une lettre de l'alphabet va être remplacée par une autre. La version la plus communément utilisée est de remplacer une lettre par la lettre se situant trois lettres à sa gauche. Par exemple, un D deviendra un A, et un E deviendra un B. Ce procédé est illustré sur la figure suivante :

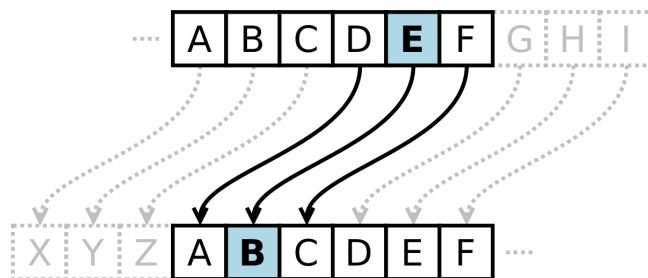


FIGURE 1 – Illustration du chiffrement de César (source : Wikipedia)

Attention : il y a trois cas spéciaux, qui sont A, B, et C qui deviendront X, Y, et Z respectivement.

Pour simplifier l'exercice, on considère que les seules lettres utilisées sont les lettres minuscules et qu'il n'y a pas d'espace entre les mots.

Vous devrez utiliser deux fonctions pour pouvoir réussir les questions suivantes :

- La fonction *chr*, qui vous donnera le nombre correspondant au code de la lettre. Par exemple, *chr('a')* retournera la valeur 0, *chr('b')* retournera la valeur 1, et *chr('z')* retournera la valeur 25.
- La fonction *ord*, qui vous donnera la lettre correspondant au nombre donné. Par exemple, *ord(0)* retournera la valeur 'a', *ord(1)* retournera la valeur 'b', et *ord(25)* retournera la valeur 'z'.

Questions :

1. Ecrire la fonction *encrypt* qui chiffrera une chaîne de caractère avec le chiffrement de César qui utilisera un décalage de trois lettres vers la gauche (le même que sur la figure 1).
2. Ecrire la fonction *decrypt* qui déchiffrera une chaîne de caractère chiffrée avec l'algorithme de la question précédente.
3. Réécrivez maintenant les deux fonctions précédentes pour un nombre de décalage quelconque (qui ira donc de 1 à 25).
4. (Bonus) Le chiffrement de César est trop simple pour être utilisé pour chiffrer les communication d'aujourd'hui. Comment feriez-vous pour déchiffrer un message chiffré avec le chiffrement de César sans savoir quel décalage a été utilisé ? Comment transformeriez-vous cette idée en code ?