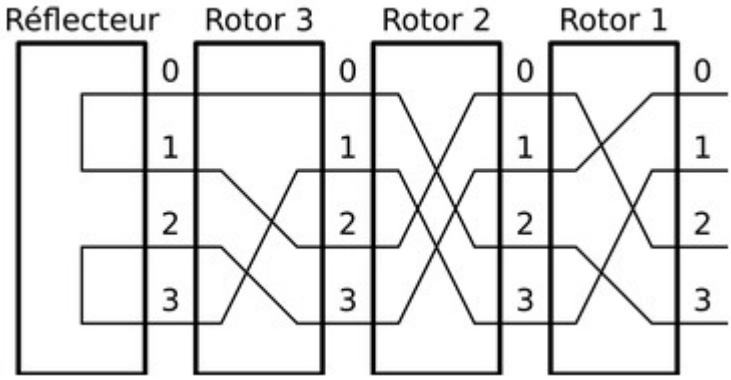
T.P. n° 11

Le but de ce TM est de simuler la machine Enigma, utilisée par l'armée allemande pendant la seconde Guerre Mondiale pour chiffrer et déchiffrer leurs messages. Ce mécanisme de chiffrement réputé inviolable a été cassé par les Polonais puis en Angleterre par l'équipe d'Alan Turing, un des pères de l'informatique (voir le film « *Imitation Game* »).

Cette machine vise à transformer chaque lettre d'un message en une autre lettre. Chaque fois que l'utilisateur appuie sur une touche (voir les photos ci-dessous), une des lettres situées au-dessus s'allume. Pour déchiffrer, on réalise une opération similaire : on tape le message chiffré sur le clavier et les lettres du message en clair s'allument au fur et à mesure. La machine utilise des rotors (visibles sur la photo de droite) qui relient électriquement une lettre en entrée avec une lettre en sortie. Ces rotors tournent mais on considère dans un premier temps qu'ils sont fixes.

Voici un exemple avec uniquement 4 chiffres au lieu des 26 lettres pour simplifier. Le chiffre à coder est à droite (par exemple, 0). Les sorties du rotor 1 sont reliées avec les entrées du rotor 2. Les sorties du rotor 2 sont reliées aux entrées du rotor 3. Les sorties du rotor 3 sont reliées à un réflecteur qui, lui aussi, relie différemment les chiffres et renvoie sur le rotor 2 puis le rotor 1. Sur l'exemple ci-contre, le 0 en entrée est ainsi relié avec le 1. Si on appuie sur le 0, c'est le 1 qui va s'allumer. Pour déchiffrer le message, il suffit d'appuyer sur le 1 et le 0 va s'allumer puisque la machine, dans cette configuration a une connexion électrique entre le 0 et le 1.



a) On suppose qu'on utilise deux rotors seulement et qu'ils sont fixes. On représente chaque rotor par un tableau contenant les connexions de chacun des caractères dans l'ordre. Par exemple, le rotor1 de l'exemple serait représenté par {1,3,0,2} parce que ce sont ces caractères qui sont reliés respectivement à 0, 1, 2 et 3.
Voici la description de 2 rotors et d'un réflecteur, utilisant les lettres de 'a' à 'z' (sur le rotor1, le 'a' est relié à 'e', le 'b' est relié à 'l', le 'c' est relié à 'w', etc.) :

```
char[]rotor1={'e','l','w','m','h','a','s','n','o','i','x','y','b','r','z','q','d','c','v','g
','t','f','k','p','u','j'};
char[]rotor2={'d','k','s','c','l','p','b','x','w','o','t','g','u','j','f','z','e','v','r','q
','m','i','a','y','n','h'};
char[]reflec={'m','q','e','t','c','z','w','u','n','l','y','j','a','i','v','x','b','s','r','d
','h','o','g','p','k','f'};
```

Copier-coller ces lignes et écrire la fonction `decodeCar` qui étant donné les 3 tableaux et un caractère, renvoie le caractère associé. Par exemple, avec 'a', la fonction doit renvoyer 'h' (de 'a' à 'e' par le rotor 1, de 'e' à 'l' par le rotor 2, de 'l' à 'j' par le déflecteur, de 'j' à 'n' par le rotor 2 inversé et de 'n' à 'h' par le rotor 1 inversé). A l'aller, il faut juste utiliser le rang de la lettre dans l'alphabet pour trouver la lettre en sortie ; vous pouvez soustraire 'a' à une lettre pour trouver son rang ('c' – 'a' vaut 2 par exemple). Au retour, il faut chercher la lettre dans la liste pour retrouver le rang. Écrire également un programme principal pour vérifier que `decodeCar('a')` renvoie bien `'h'`.

b) Ecrire la fonction `decodeMsg` pour décoder, caractère par caractère, une chaine. Utilisez votre programme pour décoder le message : jghedvujdzydtghrvqdtgsvqhguxvs

c) La machine était réputé inviolable parce que les rotors changent après chaque lettre en décalant d'un cran les entrées et les sorties. Ainsi, après le codage de la première lettre, le rotor 1 est décalé d'un cran pour devenir :

```
{'l','w','m','h','a','s','n','o','i','x','y','b','r','z','q','d','c','v','g','t','f','k','p'
,'u','j','e'};
```

Notez bien le 'e' qui est maintenant en dernière place.

Écrire l'action `decaler` qui, étant donné un tableau de caractères représentant un rotor, décale ses éléments d'un cran vers le début, le premier élément étant affecté à la dernière case du tableau. Attention à l'ordre dans lequel vous faites ces opérations.

d) Lorsque le rotor 1 a fait un tour complet, il décale d'un cran le rotor 2, comme dans un compteur kilométrique de voiture. Modifier la fonction `decodeMsg` pour prendre en compte ces décalages.
Vous devriez alors pouvoir décoder ce message :

gvvlasgkkqycfkxbwwdqfjuzmlvkxfkvnhzjwypbqliiapbdqodgboyahvvs