

▷ **Question 1:** Concevez un algorithme qui prend en entrée une chaîne de caractères normale et qui la transforme en sa représentation T9 simplifiée (nous ne gérons ni les majuscules, ni les caractères spéciaux, ni les accents, ni les chiffres). Cet algorithme devra ensuite être implémenté en Python dans un programme qui devra se trouver dans le fichier P4.py que vous créerez.

Votre programme doit être exécutable dans un terminal via la commande suivante :

```
python3 P4.py
```

Votre programme devra, sur la base d'un fichier d'entrée dénommé E4.txt, créer un fichier de résultats R4.txt.

Le fichier E4.txt comprend plusieurs lignes (illustré ci-dessous). La première ligne vous indique combien de tests sont à réaliser. Chaque ligne suivante comprend un test sous la forme d'une chaîne de caractères à encoder en T9. Un tel fichier est illustré ci-dessous :

```
1 3
2 abba
3 bientot noel
4 vivement les vacances
```

FIGURE 6 – Exemple de fichier d'entrée pour l'exercice 3

En sortie, votre programme doit générer un fichier R4.txt qui contient :

- en première ligne : le nombre de réponses qui suivent, puis,
- une ligne par test qui contient l'encodage T9 de la chaîne initiale.

▷ **Question 2:** Concevez l'algorithme qui prend en entrée une chaîne de caractères encodée en T9 et qui renvoie la chaîne de caractères décodée.

Vous implémenterez ce programme dans un fichier nommé P4bis.py. Le programme devra, étant donné un ensemble de tests fournis dans un fichier dénommé E4bis.txt, générer un fichier R4bis.txt contenant les résultats de l'exécution.

Le fichier E4bis.txt contient :

- en première ligne : le nombre de tests, puis,
- une ligne par test qui contient une séquence T9 encodée.

Le fichier R4bis.txt devra contenir :

- en première ligne : le nombre de réponses qui suivent, puis,
- une ligne par test, contenant chacune la chaîne de caractères décodée.

★ **Exercice 5: Bonus - Chaînes rotatives (4 points)²**

Soit une chaîne de caractères de longueur n et un entier k . Vous pouvez transformer la chaîne de caractères en prenant l'un des k premiers caractères et en le mettant à la fin de la chaîne (ce qui l'enlève de sa position actuelle). Par exemple, pour $k = 1$, vous pouvez prendre la première lettre de la liste "jour" et la placer en fin de liste, ce qui vous donne "ourj". Si $k = 2$ vous pouvez choisir entre 'j' et 'o' puis mettre l'un deux en fin ce qui donnerais "ourj" ou "juro".

▷ **Question 1:** Ecrivez un programme qui calcule la plus petite chaîne suivant l'ordre lexicographique possible après un nombre illimité de mouvements (la chaîne résultant d'une rotation a toujours la même longueur que la chaîne initiale). Par exemple, la chaîne "badef" est plus petite que la chaîne "bacac" car 'c' est lexicographiquement avant 'd'.

Vous implémenterez ce programme dans un fichier nommé P5.py. Le programme devra, étant donné un ensemble de tests fournis dans un fichier dénommé E5.txt, générer un fichier R5.txt contenant les résultats de l'exécution.

Le fichier E5.txt, illustré dans la figure 7, contient :

- en première ligne : le nombre de tests, puis,
- une ligne par test qui contient : le nombre k maximum de caractères en tête de chaîne qui peuvent être sélectionnés suivi par la chaîne de caractères à traiter. Les deux éléments sont séparés par une virgule.

Le fichier R5.txt devra contenir :

- en première ligne : le nombre de réponses qui suivent, puis,
- une ligne par test qui contient la chaîne de caractères minimale (au sens lexicographique) construite pour ce test.

2. L'immense générosité de l'équipe de CS54 vous permet de cumuler jusqu'à 24 points !