

## Pour aller plus loin

Exercices inspirés de problèmes de qualification au concours Prologin (<https://prologin.org/train/>)

### 1. Exercice 1

Joseph Marchand est également un incorrigible romantique et il a écrit un poème pour son amie. Inattentif, il n'a pas remarqué que la disposition du clavier avait changé. On vous donne la disposition du clavier sur lequel il a tapé, sur lequel il pensait taper et le message écrit.

→ **Ecrire** une fonction `correcteur()` qui affiche le message qu'il a voulu écrire.

#### Contraintes

- La disposition d'un clavier est composée au maximum de 30 touches.
- Le poème est formé d'un vers (ligne).

#### Entrée

- Le premier paramètre est une chaîne de caractères représentant les touches erronées du clavier sur lequel Joseph Marchand pensait taper, dans un ordre arbitraire.
- Le deuxième paramètre est une chaîne de caractères représentant les touches réelles du clavier sur lequel Joseph Marchand a effectivement tapé, dans le même ordre que la première ligne d'entrée.
- Le troisième paramètre est le texte qui s'est affiché sur l'écran de Joseph Marchand. Ce texte ne contiendra pas d'accents, et chaque ligne contiendra moins de 80 caractères.

#### Sortie

- *N* lignes : le texte que Joseph Marchand voulait taper.

#### Commentaires

Pour plus de simplicité, les deux claviers ne différeront que par l'emplacement des lettres. Ainsi, les deux chaînes représentant les claviers ne contiendront que des lettres majuscules, et les lettres du textes d'entrée seront également en majuscule (par contre, le texte d'entrée peut contenir des symboles de ponctuation)

#### Exemples

```
>>> correcteur('QZAW', 'AWQZ', 'LE ZQGON EST HQWQRDEUX !')
'LE WAGON EST HAZARDEUX !'
>>> correcteur('ATB', 'BAT', 'LES ATBETUX SONB AETUX')
'LES BATEAUX SONT BEAUX'
```



## 2. Exercice 2

Une séquence d'ADN sera une suite finie constituée de lettres dans l'ensemble {A, T, G, C}. On cherche à analyser les fréquences d'apparition des sous-séquences d'une séquence d'ADN.

→ **Ecrire** une fonction `adn()` qui renvoie la sous-séquence contiguë de longueur `L` de la chaîne d'ADN la plus fréquente. Dans le cas où plusieurs sous-séquences de longueur apparaissent un même nombre de fois, affichez celle qui vient en premier dans l'ordre alphabétique.

### Entrée

- Le premier paramètre est la longueur `L` de la sous-séquence recherchée
- Le deuxième paramètre est une chaîne de caractères représentant la séquence d'ADN.

### Sortie

- La sous-séquence d'ADN de longueur `L` recherchée.

### Exemples

```
>>> adn(2, 'TCGTACGTAG')  
'CG'  
>>> adn(4, 'AATTCGGCCGATCGTCGAATTCGATA')  
'AATT'
```

