Exercice 1 : Élocution chez le dentiste

Chez le dentiste, la bouche grande ouverte, lorsqu'on essaie de parler, il ne reste que les voyelles. Même les ponctuations sont supprimées. Vous devez écrire une fonction dentiste qui prend en paramètre une chaîne de caractères `texte` et qui renvoie une autre chaine ne contenant que les voyelles de texte, placées dans le même ordre que dans texte.

Les voyelles sont données par :

```
VOYELLES = 'aeiouy'
```

On ne considérera que des textes écrits en minuscules, sans accents.

Compléter le programme suivant :

```
VOYELLES = "aeiouy"

texte = "Bonjour tout le monde"

resultat = ""

...
```

Exercice 2 : Validation de mot de passe

Nous utilisons tous des mots de passe pour nous connecter à différentes applications sur internet.

Choisir un bon mot de passe n'est pas si simple, celui-ci doit respecter certaines règles d'améliorer sa sécurité. On propose dans cet exercices certaines règles de validation des mots de passe.

On considère dans la suite qu'un mot de passe est *valide* s'il contient :

- au minimum 14 caractères (inclus),
- au moins une lettre majuscule,
- au moins une lettre minuscule,
- au moins un chiffre,
- au moins un caractère spécial parmi ",;:!?./§%*$£&#{}()[]-_@" (sans les guillemets).

On demande donc d'écrire un prorgamme valide qui prend une chaîne de caractères `mdp` et renvoie `True` si cette chaîne est un mot de passe valide, `False` dans le cas contraire.

On garantit que les chaînes de caractères `mdp` passées ne contiennent que des caractères alphanumériques et des caractères spéciaux.

Si chaine est une chaîne de caractères :

- `chaine.isupper()` renvoie `True` si chaine ne contient que des caractères en majuscule, `False` dans le cas contraire,
- `chaine.islower()` renvoie `True` si chaine ne contient que des caractères en minuscule, `False` dans le cas contraire,
- `chaine.isnumeric()` renvoie `True` si chaine ne contient que des chiffres, `False` dans le cas contraire.

Ces différentes méthodes sont utilisables dans le cas où chaine ne contient qu'un seul caractère.

Compléter le programme suivant :

```
SPECIAUX = ",;:!?./§%*$£&#{}()[]-_@"
mdp = "MonMotDePasse123@"
if …
```

Exercice 3 : Mot à trous

- Un mot est ici une chaine de caractères composée uniquement de lettres de l'alphabet.
- Un mot à trous comporte également zéro, une ou plusieurs fois le caractère ".".

On dira que `mot_complet` correspond à `mot_a_trous`, si on peut remplacer chaque "." de `mot_a_trous` par une lettre de façon à obtenir `mot_complet`.

- "INFO.MA.IQUE" est un mot à trous,
- "INFORMATIQUE" est un mot qui lui correspond,
- "AUTOMATIQUE" est un mot qui ne lui correspond pas.

Écrire un programme qui affiche un booléen qui détermine si `mot_complet` correspond à `mot_a_trous`.

Compléter le programme suivant :

```
mot_complet = "maison"
mot_a_trous = "m..s.n"
if ...
```

Exercice 4 : Collage de mots

Un logiciel de reconnaissance vocale donne la liste des mots d'un candidat hésitant, il y a de nombreux « heu » que l'on souhaite supprimer de son discours.

Écrire un programme

- qui prend en entrée : une liste de mots qui sont des chaines de caractères (non vides) uniquement composées de lettres minuscules,
- et qui affiche le discours sous forme d'une seule chaine de caractères qui ne contient plus les mots "heu" de la liste. Les mots seront séparés par un seul espace. Il n'y aura pas d'espace ni au début, ni à la fin du discours.

Compléter le programme suivant :

```
mots = ["je", "heu", "pense", "heu", "que", "c'est", "bon"]
discours = ""
for …
```

Exercice 5 : Découpage de mots

Pour analyser automatiquement les débats entre des orateurs, on vous demande de créer un programme qui affiche la liste des mots de 4 lettres ou plus dans une chaine de caractères discours passée en paramètre.

On suppose que la chaine discours a été renvoyée par une fonction de reconnaissance vocale qui ignore la ponctuation et ne renvoie que des lettres minuscules sans accent ainsi que des espaces. Le discours se termine aussi par un espace.

Attention : Fonction split interdite.

Compléter le programme suivant :

```
discours = "Ceci est un exemple de texte avec quelques mots "
mots = []
mot = ""
for …
```

Exercice 6 : Page web et HTML

On dispose d'une chaine de caractères représentant une portion de code HTML et on souhaite écrire une fonction texte_brut qui supprime toutes les balises HTML de cette chaîne et renvoie le texte brut qu'il contient.

On rappelle qu'une balise HTML s'écrit sous la forme <nom_balise> ou </nom_balise>.

Si l'on considère par exemple le code html = " un exemple de chaine " alors l'appel texte_brut(html) renverra 'un exemple de chaine'.

On admet :

- qu'un caractère < déclarant une ouverture de balise est *toujours* suivi plus loin dans la chaîne d'un caractère > ;
- qu'il n'y a **jamais** de balises imbriquées.

Compléter le programme suivant :

```
html = "<p>Bonjour <b>le monde</b> !</p>"

resultat = ""

balise = False

for …
```

Exercices adaptés de CODEX : https://codex.forge.apps.education.fr/parcours/