TD 8 (programmation) - Slices et compréhensions.

Exercice 1 (Slices)

Qu'affichent les deux codes suivants ?

```python
if __name__ == "__main__":
    lst = list(range(1, 21, 2))
    print(lst[-2])
    print(lst[1:3])
    print(lst[3:1])
    print(lst[5:-1])
```

```python
if __name__ == "__main__":
    s = 'premier exercice'
    print(s[5:])
    print(s[-3:])
    print(s[:-6])
    print(s[-8:] + s[:-9])
```

Exercice 2 (Listes et ensembles en compréhension)

Qu'affiche le programme ci-dessous ?

```python
if __name__ == "__main__":
    print([i for i in range(20)])
    print([i for i in range(20) if i % 2 == 1])
    print([i * i for i in range(1, 6)])
    print({i + j for i in range(1, 6) for j in range(1, 6)})
```

Exercice 3 (Correction d'oublis)

Écrivez une fonction `voisins(mot)` d'une seule ligne qui renvoie l'ensemble des mots qu'on peut obtenir en effaçant un seul caractère du mot fourni. Par exemple, `voisins('hello')` renverra `{'hllo', 'ello', 'hell', 'helo'}`.

Exercice 4 (Couples différents)

Écrivez une fonction `couples` d'une seule ligne qui prend une liste en paramètre et renvoie l'ensemble de tous les couples de valeurs *distinctes* de la liste. Par exemple, `couples([1, 2, 3])` renverra `{(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)}` (remarquez que certains couples du résultat correspondent à la même paire d'éléments dans la liste).

Exercice 5 (Différence)

Écrivez une fonction `diff` d'une seule ligne qui prend deux listes `lst1` et `lst2` et renvoie la liste des éléments de `lst1` qui n'appartiennent pas à `lst2`. Par exemple, `diff([1, 2, 3, 4, 5], [2, 4])` renverra `[1, 3, 5]`. L'ordre original des éléments de `lst1` doit être conservé.

Exercice 6 (Sous-chaînes)

Écrivez une fonction d'une ligne qui prend une chaîne s en argument et renvoie l'ensemble de toutes les sous-chaînes de s. On ne veut pas que la chaîne vide appartienne à ce résultat. Par exemple, la fonction appliquée à `'bon'` renverra `{'b', 'o', 'n', 'bo', 'on', 'bon'}`.

Exercice 7 (Bords)

Un *bord* d'une chaîne s est une chaîne qui est à la fois préfixe et suffixe de s : par exemple, `'ab'`, `'abab'` et `'abababab'` sont les bords de `'abababab'`. Écrivez une fonction qui renvoie, en une ligne, l'ensemble de tous les bords d'une chaîne passée en paramètre.