

```
# 1. Saisie de l'utilisateur :
# Demandez à l'utilisateur de saisir une phrase.
phrase = input("Entrez une phrase : ")

# 2. Affichage de la longueur de la phrase :
# Calculez et affichez le nombre de caractères dans la phrase.
longueur_phrase = len(phrase)
print(f"La longueur de la phrase est : {longueur_phrase}")

# 3. Conversion en majuscules et minuscules :
# Affichez la phrase en majuscules.
phrase_maj = phrase.upper()
print(f"Phrase en majuscules : {phrase_maj}")

# Affichez la phrase en minuscules.
phrase_min = phrase.lower()
print(f"Phrase en minuscules : {phrase_min}")

# 4. Comptage des mots :
# Comptez le nombre de mots dans la phrase et affichez ce nombre.
# Utilisation de la méthode split() qui divise la chaîne en une liste de mots.
mots = phrase.split()
nombre_de_mots = len(mots)
print(f"Le nombre de mots dans la phrase est : {nombre_de_mots}")

# 5. Remplacement de mots :
# Demandez à l'utilisateur de saisir un mot à remplacer et un mot de remplacement.
mot_a_remplacer = input("Entrez le mot à remplacer : ")
mot_de_replacement = input("Entrez le mot de remplacement : ")

# Remplacez toutes les occurrences du mot dans la phrase par le mot de remplacement et
phrase_remplacee = phrase.replace(mot_a_remplacer, mot_de_replacement)
print(f"Nouvelle phrase : {phrase_remplacee}")

# 6. Extraction de sous-chaînes :
# Affichez les trois premiers caractères de la phrase.
trois_premiers_caracteres = phrase[:3]
print(f"Les trois premiers caractères sont : {trois_premiers_caracteres}")

# Affichez les trois derniers caractères de la phrase.
trois_derniers_caracteres = phrase[-3:]
print(f"Les trois derniers caractères sont : {trois_derniers_caracteres}")
```

```
# 7. Inversion de la phrase :
# Affichez la phrase inversée (les caractères dans l'ordre inverse).
phrase_inversee = phrase[::-1]
print(f"Phrase inversée : {phrase_inversee}")

# 8. Vérification de la présence d'un mot :
# Demandez à l'utilisateur de saisir un mot.
mot_a_rechercher = input("Entrez un mot à rechercher : ")

# Vérifiez si ce mot est présent dans la phrase et affichez un message approprié.
if mot_a_rechercher in phrase:
    print(f"Le mot '{mot_a_rechercher}' est présent dans la phrase.")
else:
    print(f"Le mot '{mot_a_rechercher}' n'est pas présent dans la phrase.")
```

Explication détaillée :

1. Saisie de l'utilisateur :

```
phrase = input("Entrez une phrase : ")
```

- Utilisation de la fonction `input()` pour demander à l'utilisateur de saisir une phrase.
- La phrase saisie est stockée dans la variable `phrase`.

2. Affichage de la longueur de la phrase :

```
longueur_phrase = len(phrase)
print(f"La longueur de la phrase est : {longueur_phrase}")
```

- Utilisation de la fonction `len()` pour calculer le nombre de caractères dans la phrase.
- Affichage de la longueur de la phrase.

3. Conversion en majuscules et minuscules :

```
phrase_maj = phrase.upper()
print(f"Phrase en majuscules : {phrase_maj}")
```

- Utilisation de la méthode `upper()` pour convertir la phrase en majuscules.
- Affichage de la phrase en majuscules.

```
phrase_min = phrase.lower()
print(f"Phrase en minuscules : {phrase_min}")
```

- Utilisation de la méthode `lower()` pour convertir la phrase en minuscules.
- Affichage de la phrase en minuscules.

4. Comptage des mots :

```
mots = phrase.split()
nombre_de_mots = len(mots)
print(f"Le nombre de mots dans la phrase est : {nombre_de_mots}")
```

- Utilisation de la méthode `split()` pour diviser la phrase en une liste de mots.
- Utilisation de la fonction `len()` pour compter le nombre de mots dans la liste.
- Affichage du nombre de mots.

5. Remplacement de mots :

```
mot_a_remplacer = input("Entrez le mot à remplacer : ")
mot_de_replacement = input("Entrez le mot de remplacement : ")
```

- Utilisation de la fonction `input()` pour demander à l'utilisateur de saisir le mot à remplacer et le mot de remplacement.

```
phrase_remplacee = phrase.replace(mot_a_remplacer, mot_de_replacement)
print(f"Nouvelle phrase : {phrase_remplacee}")
```

- Utilisation de la méthode `replace()` pour remplacer toutes les occurrences du mot dans la phrase par le mot de remplacement.
- Affichage de la nouvelle phrase.

6. Extraction de sous-chaînes :

```
trois_premiers_caracteres = phrase[:3]
print(f"Les trois premiers caractères sont : {trois_premiers_caracteres}")
```

- Utilisation de la syntaxe de découpage `[:3]` pour extraire les trois premiers caractères de la phrase.
- Affichage des trois premiers caractères.

```
trois_derniers_caracteres = phrase[-3:]
print(f"Les trois derniers caractères sont : {trois_derniers_caracteres}")
```

- Utilisation de la syntaxe de découpage `[-3:]` pour extraire les trois derniers caractères de la phrase.
- Affichage des trois derniers caractères.

7. Inversion de la phrase :

```
phrase_inversee = phrase[::-1]
print(f"Phrase inversée : {phrase_inversee}")
```

- Utilisation de la syntaxe de découpage `[::-1]` pour inverser la phrase.
- Affichage de la phrase inversée.

8. Vérification de la présence d'un mot :

```
mot_a_rechercher = input("Entrez un mot à rechercher : ")
```

- Utilisation de la fonction `input()` pour demander à l'utilisateur de saisir un mot à rechercher.

```
if mot_a_rechercher in phrase:
    print(f"Le mot '{mot_a_rechercher}' est présent dans la phrase.")
else:
    print(f"Le mot '{mot_a_rechercher}' n'est pas présent dans la phrase.")
```

- Utilisation de l'opérateur `in` pour vérifier si le mot est présent dans la phrase.
- Affichage d'un message approprié en fonction de la présence ou non du mot dans la phrase.