1. Reverse a String

Write a Python program to reverse a string.

print(input("Enter a string: ")[::-1])

==========

2. Check if a String is a Palindrome

Write a Python program to check if a string is a palindrome (reads the same backward as forward).

s = input("Enter a string: ")

print(s == s[::-1])

==========

3.Remove Duplicates from a String

Write a Python program to remove duplicate characters from a string.

s="hello"

print("".join(set(s)))

==========

4.Find the Longest Word in a String

Write a Python program to find the longest word in a given string.

text = "Python is a great programming language"

#Output=programming

text = "Python is a great programming language"

print(max(text.split(), key=len))

==========

5.Find Common Elements Between Two Tuples

Write a Python program to find common elements between two tuples.

``` python

tuple1 = (1, 2, 3)

tuple2 = (2, 3, 4)

# Output: (2, 3)

print(tuple(set(tuple1) & set(tuple2)))

print(tuple(x for x in tuple1 if x in tuple2))
```

==========

6.Find the Maximum and Minimum Value in a Dictionary

Write a Python program to find the maximum and minimum value in a dictionary.

``` Python

my_dict = {"a": 10, "b": 20, "c": 5}

//Min= 5  , max=20

print("Min=",min(list(my_dict.values())),"  ,Max=",max(list(my_dict.values())) )
```

==========

7- Merge Two Dictionaries

Write a Python program to merge two dictionaries.

```
dict1 = {"a": 1, "b": 2}

dict2 = {"c": 3, "d": 4}

print(dict1.update(dict2))

print(dict1)
```

==========

8- Find Common Keys in Two Dictionaries

Write a Python program to find common keys in two dictionaries.

```
dict1 = {"a": 1, "b": 2, "c": 3}

dict2 = {"b": 2, "c": 4, "d": 5}

#Output: {'b', 'c'}

print(set(dict1.keys() & dict2.keys()))
```

==========

9- takes a string and prints the longest

alphabetical ordered substring occurred.

For example, if the string is 'abdulrahman' then the output is:

Longest substring in alphabetical order is: abdu

```
 s = input("Enter a string: ")
longest = current = s[0]
for i in range(1, len(s)):
   if s[i] >= s[i - 1]:
      current += s[i]
      if len(current) > len(longest):
         longest = current
   else:
      current = s[i]
print("Longest substring:", longest)
```