Assignment -6
Python programming

1. Write a Python program to check if a number is positive, negative or zero.

Code:

```
a=int(input("Enter A: "))

if(a>0):
   print("Number is positive")
elif(a<0):
   print("Number is negative")
elif(a==0):
   print("Zero")
```

Output:

```
>>>
    ===== RESTART: C:/Users/user/AppData/Local/Programs/Python/
    Enter A: -1
    Number is negative
>>>
    ===== RESTART: C:/Users/user/AppData/Local/Programs/Python
    Enter A: 4
    Number is positive
>>>
>>>
    ===== RESTART: C:/Users/user/AppData/Local/Programs/Python/
    Enter A: 0
    Zero
>>>
```

2. Write a Python program to get the Factorial number of given number.

Code:

```
num = int(input("Enter a number :"))
factorial = 1
if num<0:
   print("factorial doen not exist for negative number")
elif num==0:
   print("the factorial of 0 is 1")
else:
   for i in range(1,num+1):
      factorial=factorial*i
   print("the factorial of",num,"is",factorial)
```

Output:

```
>>>
===== RESTART: C:/Users/user/AppData/Local/Programs/Python
Enter a number :6
the factorial of 6 is 720
>>>
```

3. Write a Python program to get the Fibonacci series of given range.

Code:

```
n=10
num1=0
num2=1
next_number=num2
count=1

while count<=n:
   print(next_number, end="")
   count+=1
   num1,num2=num2,next_number
   next_number=num1+num2
   print()
```

Output:

```
===== RESTART: C:/Users/user/AppData/Local/Progr
1
2
3
5
8
13
21
34
55
89
>>>
```

4. How memory is managed in Python?

-> Python uses the dynamic memory allocation which is managed by the Heap data structure. Memory Heap holds the objects and other data structures that will be used in the program.

5. What is the purpose continue statement in python?

-> The continue statement is used to skip the remaining code inside a loop for the current iteration only.

6. Write python program that swap two number with temp variable and without temp variable.

Code:

```
x=input("enter value of x:")
y=input("enter value of y:")

temp=x
x=y
y=temp

print("the value of x after swapping:{}".format(x))
print("the value od y after swapping:{}".format(y))
```

Output:

```
===== RESTART: C:/Users/user/AppData/Local/Prog
enter value of x:7
enter value of y:9
the value of x after swapping:9
the value od y after swapping:7
>>>
```

7. Write a Python program to find whether a given number is even or odd, print out an appropriate message to the user.

Code:

```
num=int(input("Enter a number:"))

if(num%2)==0:
   print("even number")
else:
   print("odd number")
```

Output:

```
===== RESTART: C:/Users/user/AppData/Local/Programs
Enter a number:5
odd number
>>>
===== RESTART: C:/Users/user/AppData/Local/Programs/
Enter a number:2
even number
>>>
```

8. Write a Python program to test whether a passed letter is a vowel or not.

Code:

```
character = input("Enter a character: ")

vowels = ['a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U']

if character in vowels:
   print("The character is a vowel")
else:
   print("The character  is a consonant")
```

Output:

```
===== RESTART: C:/Users/user/AppData/Local/Programs/P
Enter a character: w
The character  is a consonant
>>>
===== RESTART: C:/Users/user/AppData/Local/Programs/
Enter a character: a
The character character is a vowel
>>
```

9. Write a Python program to sum of three given integers. However, if two values are equal sum will be zero.

Code:

```
def sum(x,y,z):
   if x==y or y==z or z==x:
      sum=0
   else:
      sum=x+y+z
   return sum
print(sum(2,2,2))
print(sum(2,3,2))
print(sum(3,4,5))
```

Output:

```
===== RESTART: C:/Users/user/AppData/Local/Programs/Py
0
0
12
>>>
```

10. Write a Python program that will return true if the two given integer values are equal or their sum or difference is 5.

Code:

```
def test_number5(x, y):
   if x == y or abs(x - y) == 5 or (x + y) == 5:
      return True
   else:
      return False
print(test_number5(7, 2))
print(test_number5(3, 2))
print(test_number5(2, 2))
print(test_number5(7, 3))
print(test_number5(27, 53))
```

Output:

```
===== RESTART: C:/Users/user/AppData/Local/Programs,
True
True
True
False
False
>>>
```

11. Write a python program to sum of the first n positive integers.

Code:

```
n=int(input("enter a number:"))

sum_num=(n*(n+1))/2

print("sum of the first",n,"positive integer:",sum_num)
```

Output:

```
>>>
===== RESTART: C:/Users/user/AppData/Local/Programs,
enter a number:6
sum of the first 6 positive integer: 21.0
>>>
```

12. Write a Python program to calculate the length of a string.

Code:

```
str=input("enter a string:")

counter=0
for s in str:
   counter=counter+1
print("length of the input string is:",counter)
```

Output:

```
===== RESTART: C:/Users/user/AppData/Local/Programs,
enter a string:unnati
length of the input string is: 6
>>>
```

13. Write a Python program to count the number of characters (character frequency) in a string.

Code:

```
string=input("Enter a string: ")

for i in string:
   frequency = string.count(i)
   print(str(i) + ": " + str(frequency), end=", ")
```

Output:

```
===== RESTART: C:/Users/user/AppData/Local/Programs/Python/Python310/
Enter a string: unnatipargi
u: 1, n: 2, n: 2, a: 2, t: 1, i: 2, p: 1, a: 2, r: 1, g: 1, i: 2,
>>>
```

14. What are negative indexes and why are they used?

-> Negative indexing is used in Python to manipulate sequence objects such as lists, arrays, strings.

15. Write a Python program to count occurrences of a substring in a string.

Code:

```
s = 'python is popular programming language'

print('Number of occurrence of p:', s.count('p'))
```

Output:

```
============ RESTART: C:/Users/user/OneDrive/Documents
Number of occurrence of p: 4
>>>
```

16. Write a Python program to count the occurrences of each word in a given sentence

Code:

```
def word_count(str):
   counts = dict()
   words = str.split()

   for word in words:
      if word in counts:
         counts[word] += 1
      else:
         counts[word] = 1
   return counts

print( word_count('the quick brown fox jumps over the lazy dog.'))
```

Output:

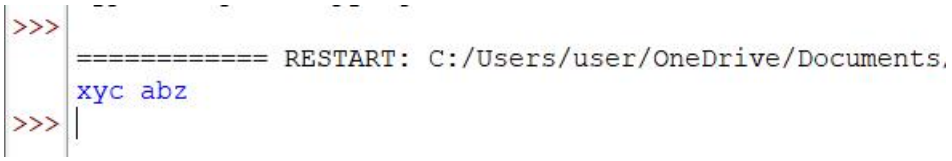
```
============ RESTART: C:/Users/user/OneDrive/Documents/Python/p8.py ============
{'the': 2, 'quick': 1, 'brown': 1, 'fox': 1, 'jumps': 1, 'over': 1, 'lazy': 1, '
dog.': 1}
>>>
```

17. Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.

Code:

```
def chars_mix_up(a,b):
   new_a = b[:2] + a[2:]
   new_b = a[:2] + b[2:]
   return new_a+ ' '+new_b
print(chars_mix_up('abc','xyz'))
```

Output:



18. Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead if the string length of the given string is less than 3, leave it unchanged.

Code:

```
def add_string(str1):
   length = len(str1)
   if length > 2:
      if str1[-3:] == 'ing':
         str1 += 'ly'
      else:
         str1 += 'ing'
      return str1

print(add_string('abc'))
print(add_string('string'))
```

Output:

```
>>>
============ RESTART: C:/Users/user/OneDrive/
abcing
stringly
>>>
```

19. Write a Python program to find the first appearance of the substring 'not' and 'poor' from a given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string.

Code:

```
def not_poor(str1):
   snot = str1.find('not')
   spoor = str1.find('poor')
   if spoor > snot and snot > 0 and spoor > 0:
      str1 = str1.replace(str1[snot:(spoor+4)],'good')
      return str1
   else:
      return str1
print(not_poor('The lyrics is not that poor'))
print(not_poor('The lyrics is poor'))
```

Output:

```
>>>
============ RESTART: C:/Users/user/OneDrive/
The lyrics is good
The lyrics is poor
>>>
```

20. Write a Python function that takes a list of words and returns the length of the longest one.

Code:

```
a=[]
n= int(input("Enter the number of elements in list:"))
for x in range(0,n):
   element=input("Enter element" + str(x+1) + ":")
   a.append(element)
max1=len(a[0])
temp=a[0]
for i in a:
   if(len(i)>max1):
      max1=len(i)
      temp=i
print("The word with the longest length is:")
print(temp)
```

Output:

```
>>>
============ RESTART: C:/Users/user/OneDrive/Doc
Enter the number of elements in list:4
Enter element1:unnati
Enter element2:jay
Enter element3:komal
Enter element4:nikhil
The word with the longest length is:
unnati
```

21. Write a Python function to reverses a string if its length is a multiple of 4.

Code:

```
str=input("Enter the String :")
if len(str) % 4 == 0:
  print(''.join(reversed(str)))
else:
        print(str)
```

Output:

```
>>>
============ RESTART: C:/Users/user/OneDrive/
Enter the String :unnati
unnati
>>>

>>>
============ RESTART: C:/Users/user/OneDrive
Enter the String :computer
retupmoc
>>>
```

22. Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.

Code:

```
def string_both_ends(str):
   if len(str) < 2:
      return ''
   return str[0:2] + str[-2:]
print(string_both_ends('unnatipargi'))
print(string_both_ends('un'))
print(string_both_ends('u'))
```

Output:

```
>>>
============ RESTART: C:/Users/user/OneDrive/
ungi
unun

>>>
```

23. Write a Python function to insert a string in the middle of a string.

Code:

```
def insert_in_middle(original_str, to_insert):

    middle_index = len(original_str) // 2

    result_str = original_str[:middle_index] + to_insert + original_str[middle_index:]

    return result_str

original_string = "welcom to    programming"
string_to_insert = "Python"
result = insert_in_middle(original_string, string_to_insert)
print(result)
```

Output:

```
>>>
============ RESTART: C:/Users/user/Onel
welcom to   Python programming
>>>
```