KUMARAGURU college of technology
character is life

Roll No: 20BMC046

Name: Vibin R

I YEAR BE / BTech

U18CSI2201 – Python Programming

Assignment I

Total Marks:30

I. Write the output for the following: (1x5=5 Marks)

1. Apply the given operations on the given string and write the output:

mystring= "python is fun"

- len(mystring)
- mystring [: : -1]
- mystring [9]
- mystring[-3]
- mystring [0:len(mystring):2]
- mystring [4:9]
- mystring [::]
- mystring [5:]

Code:

```
#Vibin_20BMC046

mystring= "python is fun"
print(len(mystring))
print(mystring [: : -1])
print(mystring [9])
print(mystring[-3])
print(mystring [0:len(mystring):2])
print(mystring [4:9])
print(mystring [::])
print(mystring [5:])
```

Output:

```
In [10]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')
13
nuf si nohtyp

f
pto sfn
on is
python is fun
n is fun
```

2.

```
#Vibin_20BMC046
msg="amichoksi"
for i in msg:
    print(i)
```

Output:

```
In [11]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')
a
m
i
c
h
o
k
s
i
```

3.

```
#Vibin_20BMC046
st="Hello world, How are you"
rt=st.replace("H","J")
print("Original String : ",st)
print("Replaced String :",rt)
```

Output:

```
In [12]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')
Original String :  Hello world, How are you
Replaced String : Jello world, Jow are you
```

4.

```
#Vibin_20BMC046
str1 = '{2}, {1} and {0}'.format('a', 'b', 'c')
str2 = '{0}{1}{0}'.format('hi', 'hello')
print(str1, str2)
```

Output:

```
In [1]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')
c, b and a hihellohi
```

5.

```
#Vibin_20BMC046
line =  "What will have so will"
L = line.split('a')
for i in L:
    print(i, end=' ')
```

Output:

```
In [2]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')
Wh t will h ve so will
```

II. **Implement the following using Python: (5x5=25 marks)**

1.Write a program that takes an IP address of the form P.Q.R.S as input, where P, Q, R and S are decimal numbers in the range 0 to 255, and prints the class of the address as indicated in the table below.

Value of P	Class
1 – 126	A
128 – 191	B
192 – 223	C
224 – 239	D
240 – 254	E

Test Case	1	2	3	4	5
Input	224.220.206.91	126.220.206.91	127.0.0.1	0.100.100.100	255.255.255.255
Output	Class D	Class A	Invalid	Invalid	Invalid

Code:

```
#Vibin_20BMC046
ip=input("Enter the IP address in the form P.Q.R.S:")
L=ip.split(".")
f=0
for i in L:
   if int(i)<1 or int(i)>224:
      f=1
      break
a=int(L[0])
if f==1:
   print("Invalid")
else:
   if a>0 and a<127:
      print("Class A")
   elif a>126 and a<192:
      print("Class B")
   elif a>191 and a<224:
      print("Class C")
   elif a>223 and a<240:
      print("Class D")
   elif a>239 and a<255:
      print("Class E")
```

Output:

```
In [1]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter the IP address in the form P.Q.R.S:224.220.206.91
Class D

In [2]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter the IP address in the form P.Q.R.S:126.220.206.91
Class A

In [3]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter the IP address in the form P.Q.R.S:127.0.0.1
Invalid

In [4]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter the IP address in the form P.Q.R.S:0.100.100.100
Invalid

In [5]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter the IP address in the form P.Q.R.S:255.255.255.255
Invalid
```

---

2. Write a program that prompts the user to enter a list of words and stores them in a list. Create a new list that retrieves words from the first list such that first letter occurs again within the word. The program should display the resulting list.

Test case	**1**	**2**
Input	Baboon, List, Duplicate	Frog, Snake, Lizard
Output	Baboon	No such word exist in list

Code:

```
#Vibin_20BMC046
List=input("Enter a list of words seperated by Commas:").split(",")
print("Entered list of words:\n",List)
```

```
print("Words in which first letter repeats again:")
f=[]
for s in List:
   t=s.lower()
   for i in range(1,len(t)):
      if t[i]==t[0]:
         f.append(s)
         break
if len(f)==0:
   print("No such words in List")
else:
   print(f)
```

Output:

```
In [8]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter a list of words seperated by Commas:Baboon,List,Duplicate
Entered list of words: ['Baboon', 'List', 'Duplicate']
Words in which first letter repeats again:
['Baboon']

In [9]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter a list of words seperated by Commas:Frog,Snake,Lizard
Entered list of words: ['Frog', 'Snake', 'Lizard']
Words in which first letter repeats again:
No such words in List

In [10]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter a list of words seperated by Commas:Lizard,Madam,Baboon
Entered list of words: ['Lizard', 'Madam', 'Baboon']
Words in which first letter repeats again:
['Madam', 'Baboon']
```

3. Explain with example about:
   a. Any 5 string functions
   b. String Operation

IMPORTANT: All string examples should include either your name or your roll number. General examples will NOT be considered for the evaluation of this question.

Refer: https://data-flair.training/blogs/python-string/
https://www.programiz.com/python-programming/methods/string

3 Explaining String functions and operations:

① isalnum()

This function returns true if the charecters in the string are alphanumeric and there is atleast one charecter.

Example:

```
>>> S= "VIBIN 20BMCO46"
>>> s.isalnum()   # Syntax: String_name.isalnum()
True.

>>> S='VIBIN'
>>> s.isalnum()
False
```

② isupper()

The function returns true only if all the charecters in a string are in uppercase.

Example:

```
>>> S= "Vibin"
>>> S.isupper()  # Syntax: Name.isupper().
False.
>>> S= 'VIBIN'
>>> S.isupper
True
```

③ find()

Syntax: String_Name.find("string to search")

Example:

```
>>> Str= "Vibin20BMC046"
>>> str.find ("20")
5
```

So, find() function returns the lowest index where the string mentioned in brackets starts in the string (main). It returns -1 if it doesn't find the string.

④ count()

Returns the number of occurrences of substring in main string.

Syntax: string.count(substring)

```
>>> Str = "VIBIN20BMC046"
>>> str.count ('0')
2
```

The character '0' is present 2 times.

⑤ capitalize()

It returns the copy of string with only first character capitalised.

syntax: string.capitalize()

Example:

```
>>> s= 'vibin'
>>> s.capitalize
Vibin
```

'v' is capitalized here.

String operations

① The '+' operator:

The concanteration operator '+' is used to concatenate or join two strings.

Example:

```
>>> S1 = 'Vibin'
>>> S2 = '20BMC046'
>>> S1+S2
Vibin20BMC046.
```

② The '*' Operator:

The multiplication operator (*) is used to concenhate the same string multiple times. It is also called repetition operator.

Eg:
```
>>> S2 = 'Vibin'
>>> S2 *3
Vibin Vibin Vibin
```

③ in and not in operator:

in operator is used to check whether a string is present in another string.

Eg:
```
>>> S = 'Vibin 20BMC046'
>>> "20" in S
True. # Because 20 is present in S
```

not in operator is used to check whether a string is not present.

Example:
```
>>> str = 'Vibin20BMC046'
>>> "BMC" not in S
False  # Because 'BMC' is present in str
```

④ slice operator

The slicing operator returns a substring from the main string from the specified start value, end value and step value indices.

Syntax:

String-Name [start : end : step]

Example:

```
>>> Str = "Vibin 20BMC046"
>>> Str [2:5]
bin
>>> Str [5:]   # end value not specified
20BMC046       # returns till end of string.
>>> str[1::1]
VbnOM06        # returns alternate values
>>>str[:5]
Vibin          # returns from first character
                 when start value not specified.
```

The end value given will be returned * till one less index from end value.

⑤ String comparisons:

Operators like >, <, >=, <= and != are used to compare the string based on the ASCII value of the characters in a string.

Example:

```
>>> S = 'Vibin'
>>>S1 = 'VIBIN'
>>> S1 > S
False  # because ASCII value of capital
       'I' is less than 'i'
```

4. Write a program which accepts comma separated words and prints them in alphabetic order.

Sample Input:

Enter words comma separated: cat,bat,mat,apple,orange,ant

Sample Output:

ant
apple
bat
cat
mat
orange

Code:

```
#Vibin_20BMC046
l=input("Enter words to sort separated by comma:").split(',')
l.sort()
print("The Sorted Words are:")
for i in l:
    print(i)
```

Output:

```
In [2]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter words to sort separated by comma:cat,bat,mat,apple,orange,ant
The Sorted Words are:
ant
apple
bat
cat
mat
orange

In [3]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter words to sort separated by comma:ring,hit,track,first,thirst,zest
The Sorted Words are:
first
hit
ring
thirst
track
zest
```

5. Demonstrate the following operations in list through simple examples:
(i) slice (ii) append (iii) extend (iv) index (v) pop

5 Demonstrating list operations:

(i) Slice

The slicing operator returns a subset of list, called spec slice by specifying start and end indices.

Syntax:

List_Name [Start_Index : End_Index]

Example:

```
>>> List = [10, 20, 30, 40, 50, 60, 70]
>>> List [1: 4]
20, 30, 40
```

The List will return the subset starting from index 1 to one less than end index.
Step value can also be added after end index to return an alternative second or third value

Example:

```
>>> List [1: 6 : 2]
20, 50
```

The step value skips two indices after retrieving first character.

(ii) append

append() function is used to add an element at the end of list.

Example:

```
>>> List = [1, 3, 5]        # Syntax:
>>> List.append (7)              List_name.append('object')
>>> list
[1, 3, 5, 7] → # 7 is added to end of list.
```

3) extend:

extend() function is used to append all elements of a list to another list.

Syntax:

```
List_1.extend(List_2)
```

Example:

```
>>> L1 = [10, 20, 30]
>>> L2 = [40, 60, 80]
>>> L1.extend(L2)
>>> L1
[10, 20, 30, 40, 60, 80]
```

L2 list is added to end of list L1.

4) index:

index() function returns the index of first occurance of element in the list.

Syntax:

```
List_name.index(object)
```

Example:

```
>>> L1 = [10, 20, 10, 30, 20, 40]
>>> L1.index(20)
1 # returns the first occurence of 20
```

5) pop:

pop() function removes the element from specified index and it returns it.

Syntax: List_name.pop(index)

If index is not specified then it returns and removes last element of list.

Example:

```
>>> LI = [10, 20, 30, 40]
>>> LI.pop(2)   # removes element in 2nd position
30
>>> LI.pop()   # returns last element
40.
```