Learn Complete Python In Simple Way

Topic

String Coding Interview Questions

Q1) Write a Program To REVERSE content of the given String by using slice operator?

```
input: durga
output: agrud

s = input('Enter Some String to Reverse:')
output = s[::-1]
print(output)
```

Q2) Write a Program To REVERSE content of the given String by using reversed() function?

```
input: durga
output: agrud

s=input('Enter Some String to Reverse:')
r=reversed(s)
output=''.join(r)
print(output)
```

Q3) Write a Program To REVERSE content of the given String by using while loop?

```
input: durga
output: agrud

s=input('Enter Some String to Reverse:')
output=''
i=len(s)-1
while i>=0:
    output=output+s[i]
    i=i-1
print(output)
```

Q4) Write a Program To REVERSE order of words present in the given string?

```
input: Learning Python Is Very Easy
output: Easy Very Is Python Learning

s=input('Enter Some String:')
l=s.split()
l1=l[::-1]
output=' '.join(l1)
print(output)
```

Q5) Write a Program To REVERSE internal content of each word?

```
input: 'Durga Software Solutions'
output: 'agruD erawtfoS snoituloS'

s=input('Enter Any String:')
l=s.split()
l1=[]
for word in l:
    l1.append(word[::-1])
output=' '.join(l1)
print(output)
```

Q6) Write a Program To REVERSE internal content of every second word present in the given string?

```
i/p: one two three four five six
o/p: one owt three ruof five xis

s='one two three four five six'
l=s.split()
l1=[]
i=0
while i<len(l):
    if i%2 == 0:
        l1.append(l[i])
    else:
```

```
12)       l1.append(l[i][::-1])
13)    i=i+1
14) output=' '.join(l1)
15) print(output)
```

Q7) Write a program to print the characters present at even index and odd index seperately for the given string?

1st Way:

```
1)  s=input('Enter Input String:')
2)  print('Characters present at Even Index:')
3)  i=0
4)  while i<len(s):
5)     print(s[i])
6)     i=i+2
7)  print('Characters present at Odd Index:')
8)  i=1
9)  while i<len(s):
10)    print(s[i])
11)    i=i+2
```

Output:

```
D:\durgaclasses>py test.py
Enter Input String:durgasoftware
Characters present at Even Index:
d
r
a
o
t
a
e
Characters present at Odd Index:
u
g
s
f
w
r
```

2nd Way:

```
1) s=input('Enter Input String:')
2) print('Characters present at Even Index:',s[0::2])
3) print('Characters present at Even Index:',s[::2])
4) print('Characters present at Odd Index:',s[1::2])
```

Q8) Write a program to merge characters of 2 strings into a single string by taking characters alternatively?

Input:

```
s1='RAVI'
s2='TEJA'
```

Output: RTAEVJIA

If strings are having same length:

```
1) s1='RAVI'
2) s2='TEJA'
3) output=''
4) i,j=0,0
5) while i<len(s1) or j<len(s2):
6)     output=output+s1[i]+s2[j]
7)     i=i+1
8)     j=j+1
9) print(output)
```

Output: RTAEVJIA

2nd way by using map():

```
1) s1='RAVI'
2) s2='TEJA'
3) l=list(map(lambda x,y:x+y,s1,s2))
4) print(''.join(l))
```

Note: The above program can work if the lengths of 2 strings are same.

If strings having different lengths:

```
1)  s1=input('Enter First String:')
2)  s2=input('Enter Second String:')
3)  output=''
4)  i,j=0,0
5)  while i<len(s1) or j<len(s2):
6)      if i<len(s1):
7)          output=output+s1[i]
8)          i=i+1
9)      if j<len(s2):
10)         output=output+s2[j]
11)         j=j+1
12) print(output)
```

Output:

```
D:\durgaclasses>py test.py
Enter First String:RAVIKIRAN
Enter Second String:TEJA
RTAEVJIAKIRAN

D:\durgaclasses>py test.py
Enter First String:RAVI
Enter Second String:TEJAKIRAN
RTAEVJIAKIRAN
```

Q9) Assume input string contains only alphabet symbols and digits. Write a program to sort characters of the string, first alphabet symbols followed by digits?

```
1)  input: B4A1D3
2)  output: ABD134
3)
4)  s='B4A1D3'
5)  alphabets=[]
6)  digits=[]
7)  for ch in s:
8)      if ch.isalpha():
9)          alphabets.append(ch)
10)     else:
```

```
11)       digits.append(ch)
12) output=''.join(sorted(alphabets)+sorted(digits))
13) print(output)
```

Alternative way:

```
1)  s='B4A1D3'
2)  alphabets=''
3)  digits=''
4)  for ch in s:
5)      if ch.isalpha():
6)          alphabets+=ch
7)      else:
8)          digits+=ch
9)  output=''
10) for ch in sorted(alphabets):
11)     output=output+ch
12) for ch in sorted(digits):
13)     output=output+ch
14) print(output)
```

Q10) Write a program for the following requirement?

```
1)  input: a4b3c2
2)  output: aaaabbbcc
3)
4)  s=input('Enter Some String where alphabet symbol should be followed by digit:')
5)  output=''
6)  for ch in s:
7)      if ch.isalpha():
8)          x=ch
9)      else:
10)         d=int(ch)
11)         output=output+x*d
12) print(output)
```

Q11) Write a program for the following requirement?

```
1) input: a3z2b4
2) output: aaabbbbzz  (sorted String)
3)
4) s=input('Enter Some String where alphabet symbol should be followed by digit:')
5) target=''
6) for ch in s:
7)     if ch.isalpha():
8)         x=ch
9)     else:
10)        d=int(ch)
11)        target=target+x*d
12) output = ''.join(sorted(target))
13) print(output)
```

Q12) Write a program for the following requirement?

```
1) input: aaaabbbccz
2) output: 4a3b2c1z
3)
4) s='aaaabbbccz'
5) output=''
6) previous=s[0]
7) c=1
8) i=1
9) while i<len(s):
10)    if s[i]==previous:
11)        c=c+1
12)    else:
13)        output=output+str(c)+previous
14)        previous=s[i]
15)        c=1
16)    if i ==len(s)-1:
17)        output=output+str(c)+previous
18)    i=i+1
19) print(output)
```

Q13) Write a program for the following requirement?

Input: a4k3b2
Output: aeknbd

In this example the following two functions are required to use

1) **ord(): To find unicode value for the given character**
   Eg: print(ord('a')) #97
2) **chr(): To find corresponding character for the given unicode value**
   Eg: print(chr(97)) # a

```
1)  s='a4k3b2'
2)  output=''
3)  for ch in s:
4)      if ch.isalpha():
5)          x=ch
6)          output=output+ch
7)      else:
8)          d=int(ch)
9)          newc= chr(ord(x)+d)
10)         output=output+newc
11) print(output)
```

Q14) Write a program to remove duplicate characters from the given input String?

Input: AZZZBCDABBCDABBBBCCCCDDDDEEEEEF
Output: AZBCDEF

1st way:

```
1)  s='AZZZBCDABBCDABBBBCCCCDDDDEEEEEF'
2)  output=''
3)  for ch in s:
4)      if ch not in output:
5)          output=output+ch
6)  print(output) # AZBCDEF
```

2nd way:

```
1) s='AZZZBCDABBCDABBBBCCCCDDDDEEEEEF'
2) l=[]
3) for ch in s:
4)     if ch not in l:
5)         l.append(ch)
6) output=''.join(l)
7) print(output) # AZBCDEF
```

3rd way by using set (but no guarantee for the order)

```
1) s='ABCDABXXXBCDABBBBCCCZZZZCDDDDEEEEEF'
2) s1=set(s)
3) output=''.join(s1)
4) print(output) #CAEZBFD
```

Q15) Write a program to find the number of occurrences of each character present in the given string?

By using count() method and List:

```
1) s='ABCDABXXXBCDABBBBCCCZZZZCDDDDEEEEEF'
2) l=[]
3) for ch in s:
4)     if ch not in l:
5)         l.append(ch)
6)
7) for ch in sorted(l):
8)     print('{} occurrs {} times'.format(ch,s.count(ch)))
```

Without using count() method:

```
1) s='ABCDABXXXBCDABBBBCCCZZZZCDDDDEEEEEF'
2) d={}
3) for ch in s:
4)     d[ch]=d.get(ch,0)+1
5) for k,v in d.items():
6)     print('{} occurrs {} times'.format(k,v))
```

For sorting purpose:

```
1) for k,v in sorted(d.items()):
2)     print('{} occurrs {} times'.format(k,v))
```

Q16) Write the program for the following requirement:

Input: ABAABBCA
Output: 4A3B1C

```
1) s='ABAABBCA'
2) output=''
3) d={}
4) for ch in s:
5)     d[ch]=d.get(ch,0)+1
6) for k,v in sorted(d.items()):
7)     output=output+str(v)+k
8) print(output)
```

Q17) Write the program for the following requirement:

Input: ABAABBCA
Output: A4B3C1

```
1) s='ABAABBCA'
2) output=''
3) d={}
4) for ch in s:
5)     d[ch]=d.get(ch,0)+1
6) for k,v in sorted(d.items()):
7)     output=output+k+str(v)
8) print(output)
```

Q18) Write a program to find the number of occurrences of each vowel present in the given string?

```
1) s=input('Enter some string to search for vowels:')
2) v=['a','e','i','o','u','A','E','I','O','U']
3) d={}
4) for ch in s:
5)     if ch in v:
```

```
6)        d[ch]=d.get(ch,0)+1
7)  for k,v in sorted(d.items()):
8)     print('{} occurrs {} times'.format(k,v))
```

```
D:\durgaclasses>py test.py
Enter some string to search for vowels:DURGASOFTWARESOLUTIONS
A occurrs 2 times
E occurrs 1 times
I occurrs 1 times
O occurrs 3 times
U occurrs 2 times

D:\durgaclasses>py test.py
Enter some string to search for vowels:mississippi
i occurrs 4 times
```

Q19) Write a program to check whether the given two strings are anagrams or not?

Two strings are said to be anagrams iff both are having same content irrespective of characters position.

Eg: lazy and zaly

```
1)  s1=input("Enter first string:")
2)  s2=input("Enter second string:")
3)  if(sorted(s1)==sorted(s2)):
4)      print("The strings are anagrams.")
5)  else:
6)      print("The strings aren't anagrams.")
```

Output:

```
D:\durgaclasses>py test.py
Enter first string:lazy
Enter second string:zaly
The strings are anagrams.

D:\durgaclasses>py test.py
Enter first string:durga
Enter second string:urgadd
The strings aren't anagrams.
```

Q20) Write a program to check whether the given string is palindrome or not ?

A string is said to be palindrome iff original string and its reversed strings are equal.

```
1) s=input("Enter Some string:")
2) if s==s[::-1]:
3)     print('The given string is palindrome')
4) else:
5)     print('The given string is not palindrome')
```

```
D:\durgaclasses>py test.py
Enter Some string:level
The given string is palindrome

D:\durgaclasses>py test.py
Enter Some string:madam
The given string is palindrome

D:\durgaclasses>py test.py
Enter Some string:apple
The given string is not palindrome
```

Q21) Write the program for the following requirement:

```
1) inputs:
2)     s1='abcdefg'
3)     s2='xyz'
4)     s3='12345'
5) output: ax1, by2,cz3,d4,e5,f,g
6)
7) s1='abcdefg'
8) s2='xyz'
9) s3='12345'
10) i=j=k=0
11) while i<len(s1) or j<len(s2) or k<len(s3):
12)     output=''
13)     if i<len(s1):
14)         output=output+s1[i]
15)         i=i+1
```

```
16)  if j<len(s2):
17)      output=output+s2[j]
18)      j=j+1
19)  if k<len(s3):
20)      output=output+s3[k]
21)      k=k+1
22)  print(output)
```

Output:

```
ax1
by2
cz3
d4
e5
f
g
```