

## Control Flow and Built in Data Structures

Q1)Write a code to filter all sub-strings which has even number of vowel?

```

❏ Filter_sub_strings.py > ...
1  str="I have an input string which contains even and odd numbers of vowels aA aa aaa ae aeo"
2  str=str.split()
3  vowels=['a','e','i','o','u']
4  for i in range(len(str)):
5      count=0
6      for char in str[i]:
7          if char.lower() in vowels :
8              count+=1
9          if count%2==0:
10             print(str[i])

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```

/bin/python3 "/home/rohit/Control Flow & Built in Data Structure/filter_sub_strings.py"
bash: /home/rohit/.bashrc: line 123: syntax error near unexpected token `unset'
bash: /home/rohit/.bashrc: line 123: `fi unset color_prompt force_color_prompt'
❏ rohit@TTNPL-rohitvarshney:~/Control Flow & Built in Data Structures$ /bin/python3 "/home/rohit/Control Flow & Built in Data Structure/filter_sub_strings.py"
have
input
even
numbers
vowels
aA
aa
ae
❏ rohit@TTNPL-rohitvarshney:~/Control Flow & Built in Data Structures$

```

Q2)Programming: From a multi-words and multi-line string, prepare a dict with key as "word" and value as occurrence of word.

```
◆ occurrence_words.py > ...
1 dict1={}
2
3 str="""Python Multiline String Using Triple-Quotes.
4 Using the triple quotes style is one of the easiest and most common ways to split a large string into a multiline Python string. Triple quotes (''' or ''') can
5 be used to create
6 a multiline string. It allows you to format text over many lines and include line breaks. Put two triple quotes around the multiline Python string, one at the
7 start and one at
8 the end, to define it."""
9 str=str.split()
10 for i in str:
11     if i in dict1.keys():
12         dict1[i]+=1
13     else:
14         dict1[i]=1
15
16 # for key , value in dict1.items():
17 #     print(key,"|",value)
18 print(dict1)
```

Q3)Programing: From a multi-words and multi-line string, prepare a filter the list of words which have multiple occurrences in string.

```
multiple_occurrence_words.py > ...
1 dict1={}
2 str="Python Multiline String Using Triple-Quotes.
3 Using the triple quotes style is one of the easiest and most common ways to split a large string into a multiline Python string. Triple quotes (''' or ''') can
4 be used to create
5 a multiline string. It allows you to format text over many lines and include line breaks. Put two triple quotes around the multiline Python string, one at the
6 start and one at
7 the end, to define it.""
8 str=str.split()
9 for i in str:
10     if i in dict1.keys():
11         dict1[i]+=1
12     else:
13         dict1[i]=1
14 for key in dict1.keys():
15     if dict1[key]>1:
16         print(key)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
/bin/python3 "/home/rohit/Control Flow & Built in Data Structure/multiple_occurrence_words.py"
bash: /home/rohit/.bashrc: line 123: syntax error near unexpected token `unset'
bash: /home/rohit/.bashrc: line 123: `fi unset color_prompt force_color_prompt'
rohit@TTNPL-rohitvarshney:~/Control Flow & Built in Data Structures$ /bin/python3 "/home/rohit/Control Flow & Built in Data Structure/multiple_occurrence_words.py"
Python
Using
the
triple
quotes
one
and
to
a
multiline
string.
at
rohit@TTNPL-rohitvarshney:~/Control Flow & Built in Data Structures$
```

Q4)Programing: From a multi-words and multi-line string, display list of words and word's length with occurrence more than 1 in sorted order .

```
word_length.py > ...
1 dict1={}
2 str="Python Multiline String Using Triple-Quotes.
3 Using the triple quotes style is one of the easiest and most common ways to split a large string into a multiline Python string. Triple quotes (''' or ''') can
4 be used to create
5 a multiline string. It allows you to format text over many lines and include line breaks. Put two triple quotes around the multiline Python string, one at the
6 start and one at
7 the end, to define it.""
8 str=str.split()
9 for i in str:
10     if i in dict1.keys():
11         dict1[i]+=1
12     else:
13         dict1[i]=1
14 print("Word\t", "Length\t", "Occurrence")
15 for key , value in dict1.items():
16     if value>1:
17         print(key, "\t", len(key), "\t", value)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
rohit@TTNPL-rohitvarshney:~/Control Flow & Built in Data Structures$ /bin/python3 "/home/rohit/Control Flow & Built in Data Structure/word_length.py"
Word Length Occurrence
Python 6 3
Using 5 2
the 3 5
triple 6 2
quotes 6 3
one 3 3
and 3 3
to 2 4
a 1 3
multiline 9 3
string. 7 2
at 2 2
rohit@TTNPL-rohitvarshney:~/Control Flow & Built in Data Structures$
```