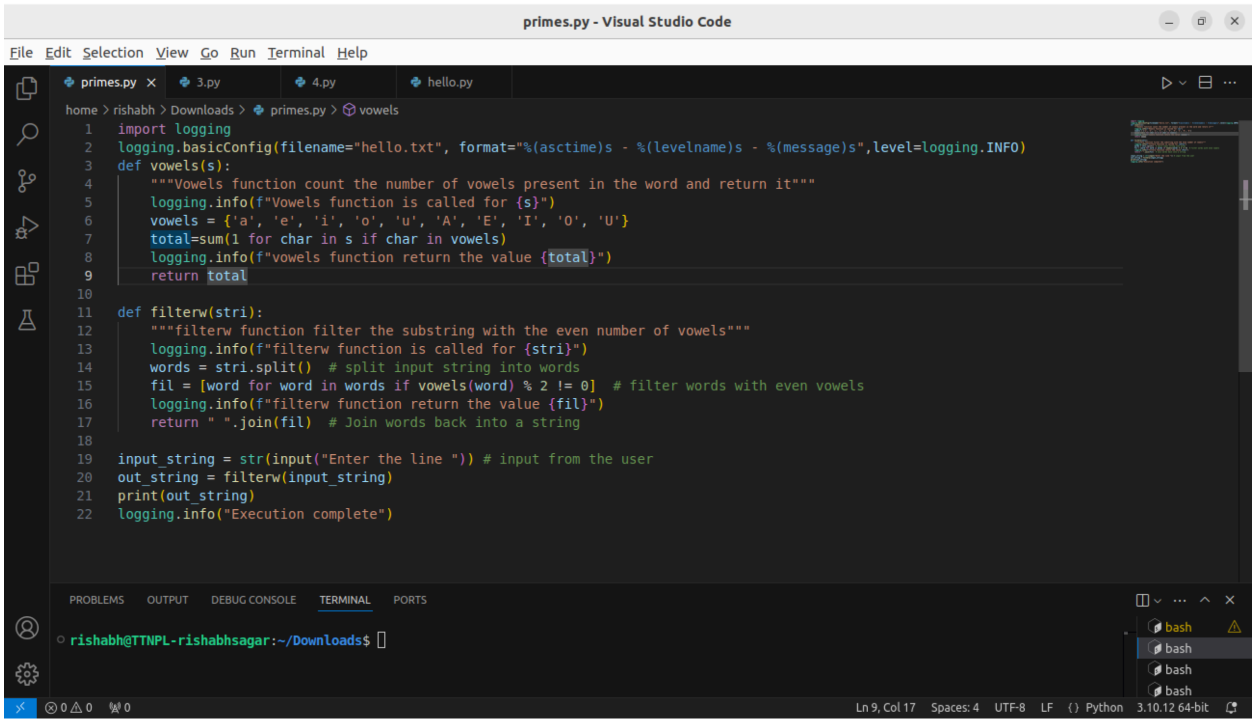
Control Flow and Built in Data Structures

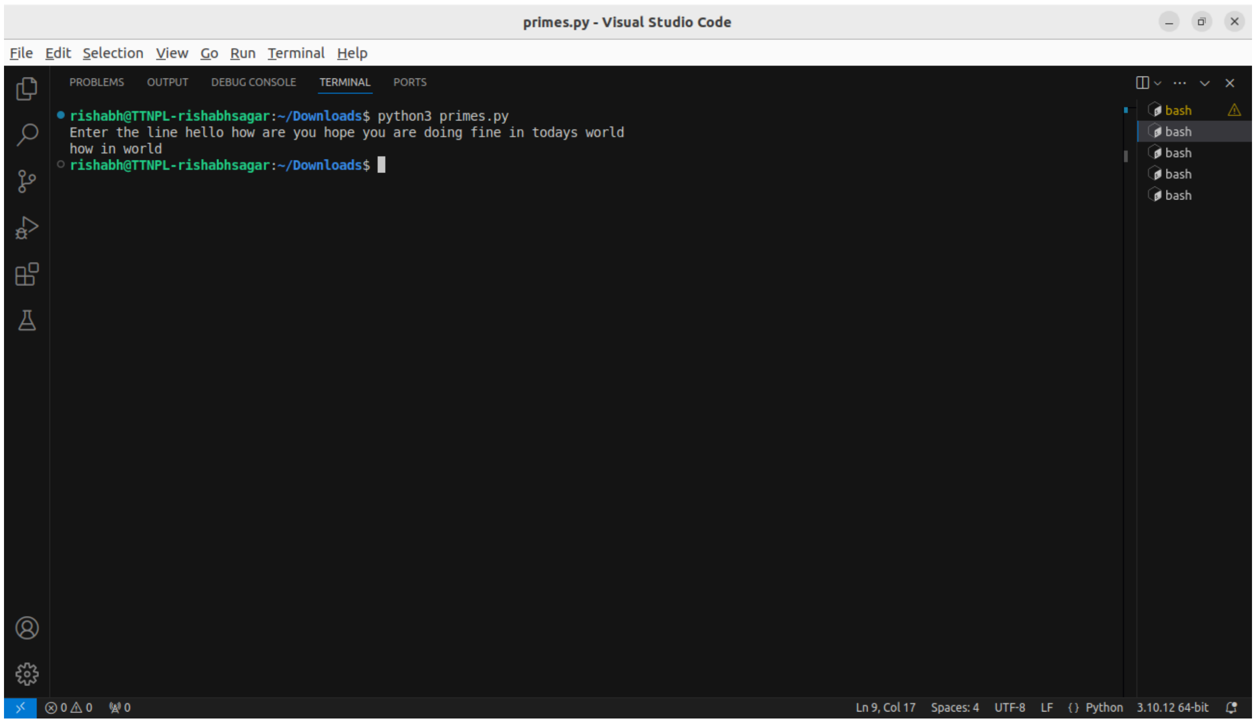
Q1)Write a code to filter all sub-strings which have an even number of vowels? example Input: "I have an input string which contains even and odd numbers of vowels aA aa aaa ae aeo” output: I an string which contains and odd of aaa aeo

Ans.

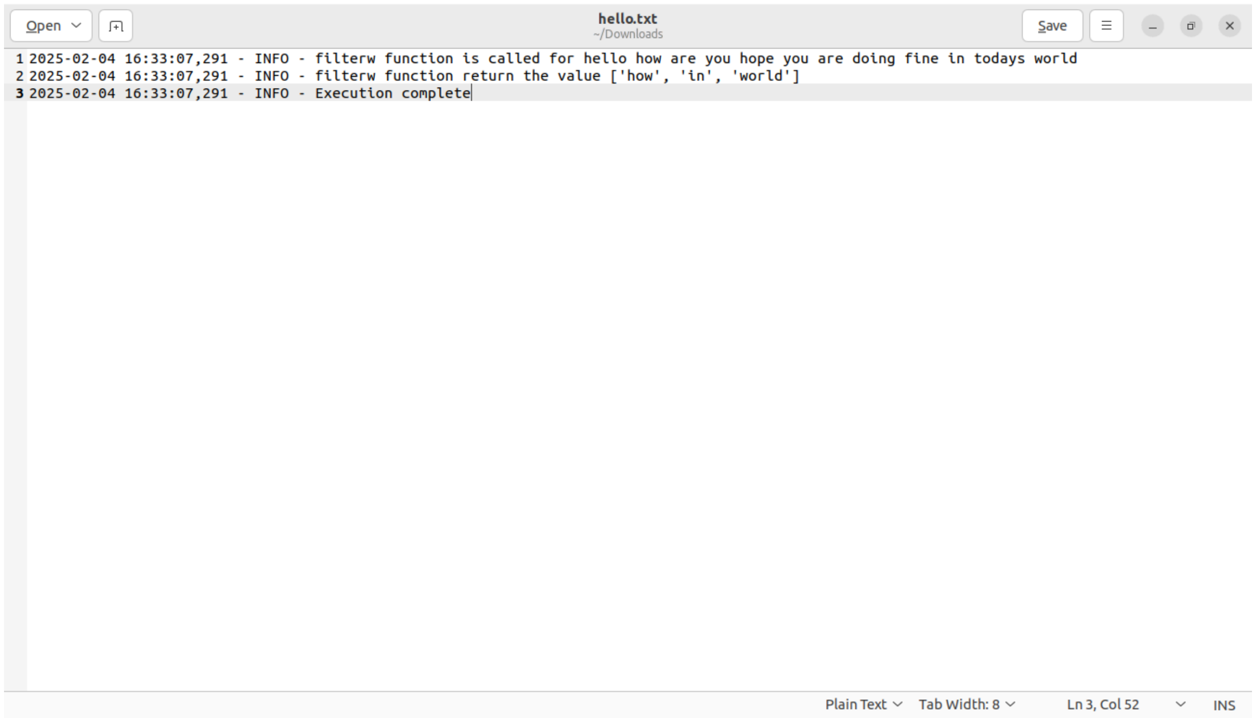

primes.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
primes.py
3.py
4.py
hello.py
home > rishabh > Downloads > primes.py > vowels
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Ln 9, Col 17 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit

```python
import logging
logging.basicConfig(filename="hello.txt", format="%(asctime)s - %(levelname)s - %(message)s",level=logging.INFO)
def vowels(s):
    """Vowels function count the number of vowels present in the word and return it"""
    logging.info(f"Vowels function is called for {s}")
    vowels = {'a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U'}
    total=sum(1 for char in s if char in vowels)
    logging.info(f"vowels function return the value {total}")
    return total

def filterw(stri):
    """filterw function filter the substring with the even number of vowels"""
    logging.info(f"filterw function is called for {stri}")
    words = stri.split()  # split input string into words
    fil = [word for word in words if vowels(word) % 2 != 0]  # filter words with even vowels
    logging.info(f"filterw function return the value {fil}")
    return " ".join(fil)  # Join words back into a string

input_string = str(input("Enter the line ")) # input from the user
out_string = filterw(input_string)
print(out_string)
logging.info("Execution complete")
```

```
rishabh@TTNPL-rishabhsagar:~/Downloads$
```


primes.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Ln 9, Col 17 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit

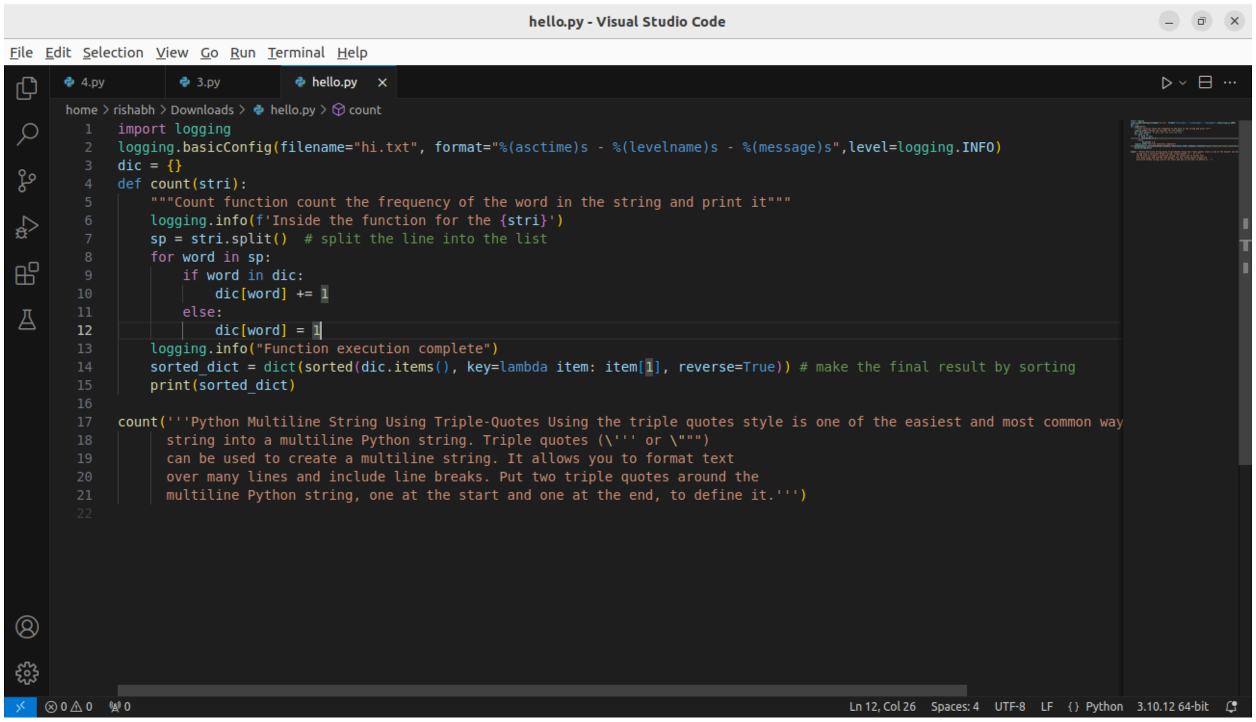
```
rishabh@TTNPL-rishabhsagar:~/Downloads$ python3 primes.py
Enter the line hello how are you hope you are doing fine in todays world
how in world
rishabh@TTNPL-rishabhsagar:~/Downloads$
```



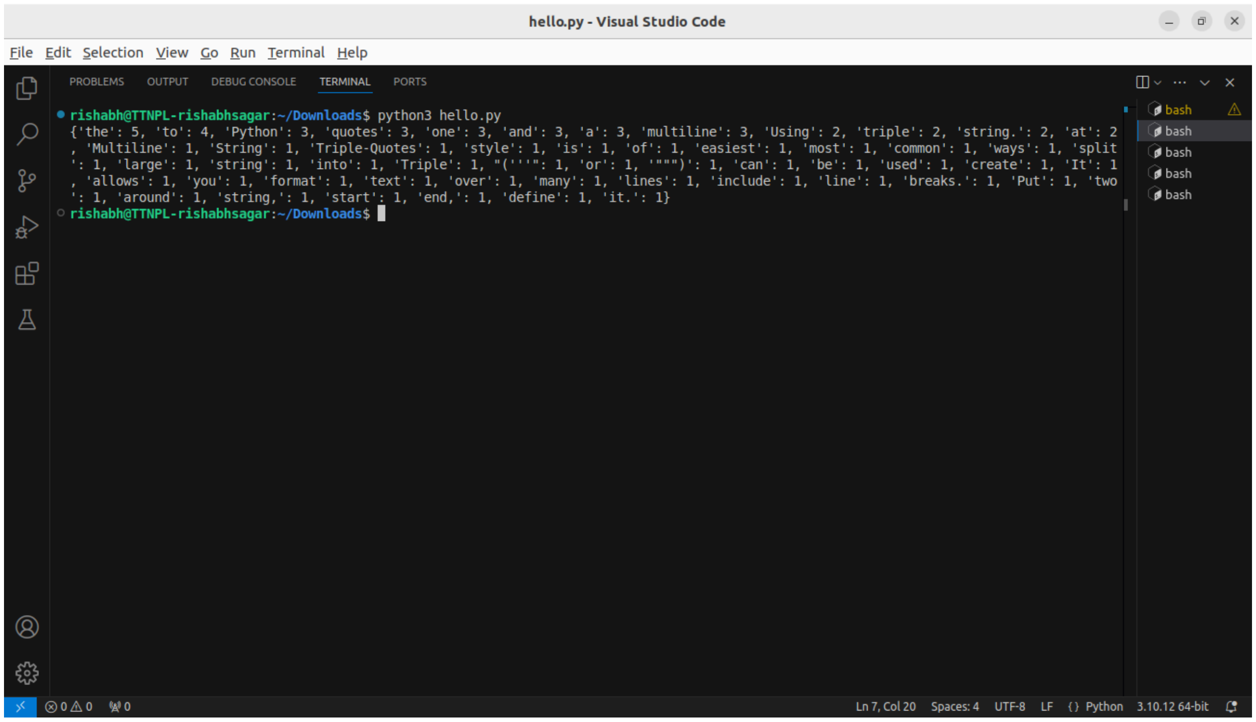
```
2025-02-04 16:33:07,291 - INFO - filterw function is called for hello how are you hope you are doing fine in todays world
2025-02-04 16:33:07,291 - INFO - filterw function return the value ['how', 'in', 'world']
2025-02-04 16:33:07,291 - INFO - Execution complete
```

Q2)Programing: From a multi-words and multi-line string, prepare a dict with key as "word" and value as occureance of word. Example Input: astring = """Python Multiline String Using Triple-Quotes 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 be used to create 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 start and one at the end, to define it.""" output (Partical): {'the': 5, 'to': 4, 'Python': 3, 'quotes': 3, 'one': 3, 'and': 3, 'a': 3, 'multiline': 3, 'Using': 2, 'triple': 2, 'string.': 2, 'at': 2, 'Multiline': 1, 'String': 1, 'Triple-Quotes': 1, 'style': 1, 'is': 1, 'of': 1, 'easiest': 1, 'most': 1, 'common': 1, 'ways': 1, 'split': 1, 'large': 1, 'string': 1, 'into': 1, 'Triple': 1, "('''": 1, 'or': 1, '""")': 1, 'can': 1, 'be': 1, 'used': 1, 'create': 1, 'It': 1, 'allows': 1, 'you': 1, 'format': 1, 'text': 1, 'over': 1, 'many': 1, 'lines': 1, 'include': 1, 'line': 1, 'breaks.': 1, 'Put': 1, 'two': 1, 'around': 1, 'string,': 1, 'start': 1, 'end,': 1, 'define': 1, 'it.': 1}

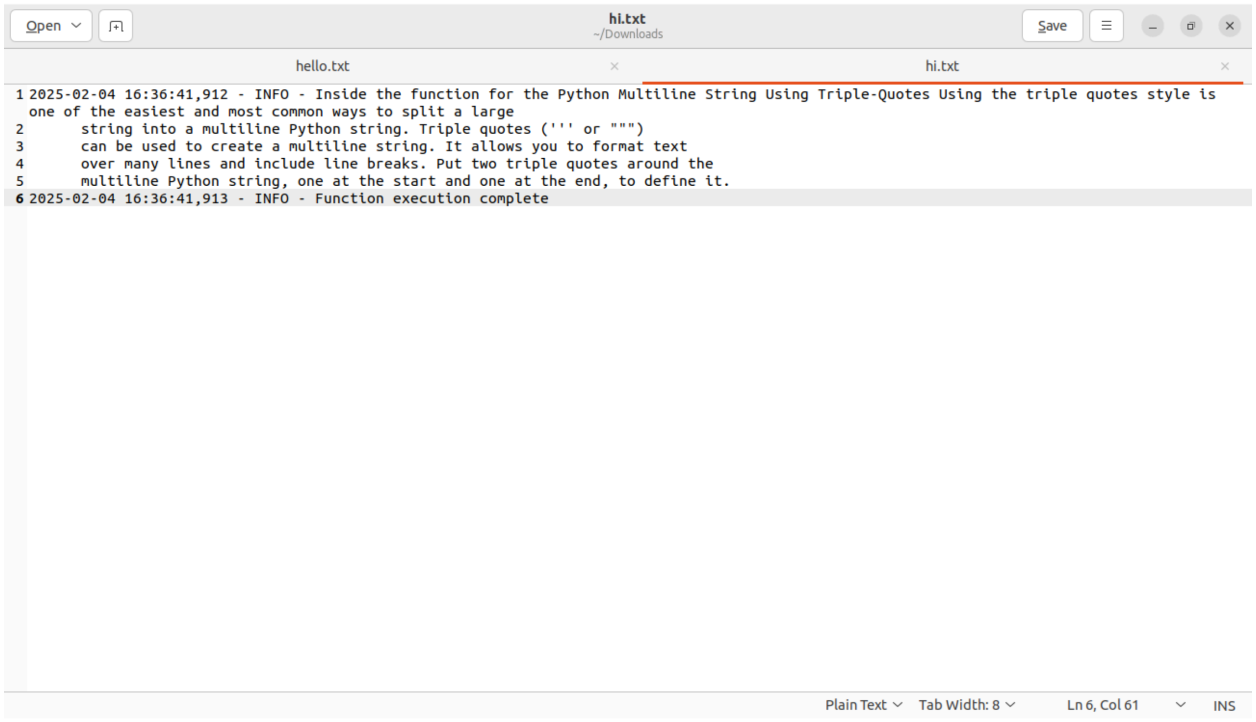
Ans.


hello.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
4.py
3.py
hello.py
home > rishabh > Downloads > hello.py > count
Ln 12, Col 26
Spaces: 4
UTF-8
LF
Python
3.10.12 64-bit

```python
import logging
logging.basicConfig(filename="hi.txt", format="%(asctime)s - %(levelname)s - %(message)s",level=logging.INFO)
dic = {}
def count(stri):
    """Count function count the frequency of the word in the string and print it"""
    logging.info(f'Inside the function for the {stri}')
    sp = stri.split()  # split the line into the list
    for word in sp:
        if word in dic:
            dic[word] += 1
        else:
            dic[word] = 1
    logging.info("Function execution complete")
    sorted_dict = dict(sorted(dic.items(), key=lambda item: item[1], reverse=True)) # make the final result by sorting
    print(sorted_dict)

count('''Python Multiline String Using Triple-Quotes Using the triple quotes style is one of the easiest and most common way
      string into a multiline Python string. Triple quotes (\''' or \""")
      can be used to create 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 start and one at the end, to define it.''')
```



```
rishabh@TTNPL-rishabhsagar:~/Downloads$ python3 hello.py
{'the': 5, 'to': 4, 'Python': 3, 'quotes': 3, 'one': 3, 'and': 3, 'a': 3, 'multiline': 3, 'Using': 2, 'triple': 2, 'string.': 2, 'at': 2
, 'Multiline': 1, 'String': 1, 'Triple-Quotes': 1, 'style': 1, 'is': 1, 'of': 1, 'easiest': 1, 'most': 1, 'common': 1, 'ways': 1, 'split
': 1, 'large': 1, 'string': 1, 'into': 1, 'Triple': 1, "('''": 1, 'or': 1, '""")': 1, 'can': 1, 'be': 1, 'used': 1, 'create': 1, 'It': 1
, 'allows': 1, 'you': 1, 'format': 1, 'text': 1, 'over': 1, 'many': 1, 'lines': 1, 'include': 1, 'line': 1, 'breaks.': 1, 'Put': 1, 'two
': 1, 'around': 1, 'string,': 1, 'start': 1, 'end,': 1, 'define': 1, 'it.': 1}
rishabh@TTNPL-rishabhsagar:~/Downloads$
```

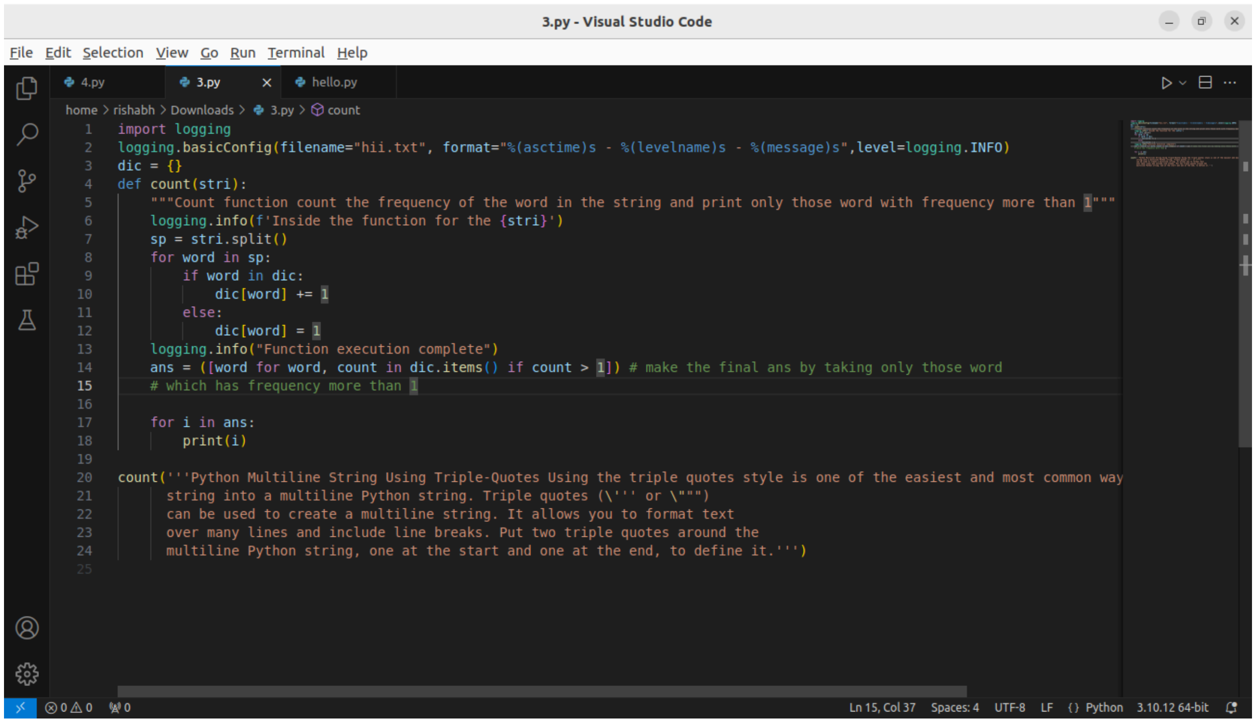


```
2025-02-04 16:36:41,912 - INFO - Inside the function for the Python Multiline String Using Triple-Quotes Using the triple quotes style is one of the easiest and most common ways to split a large
      string into a multiline Python string. Triple quotes (''' or """)
      can be used to create a 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 start and one at the end, to define it.
2025-02-04 16:36:41,913 - INFO - Function execution complete
```

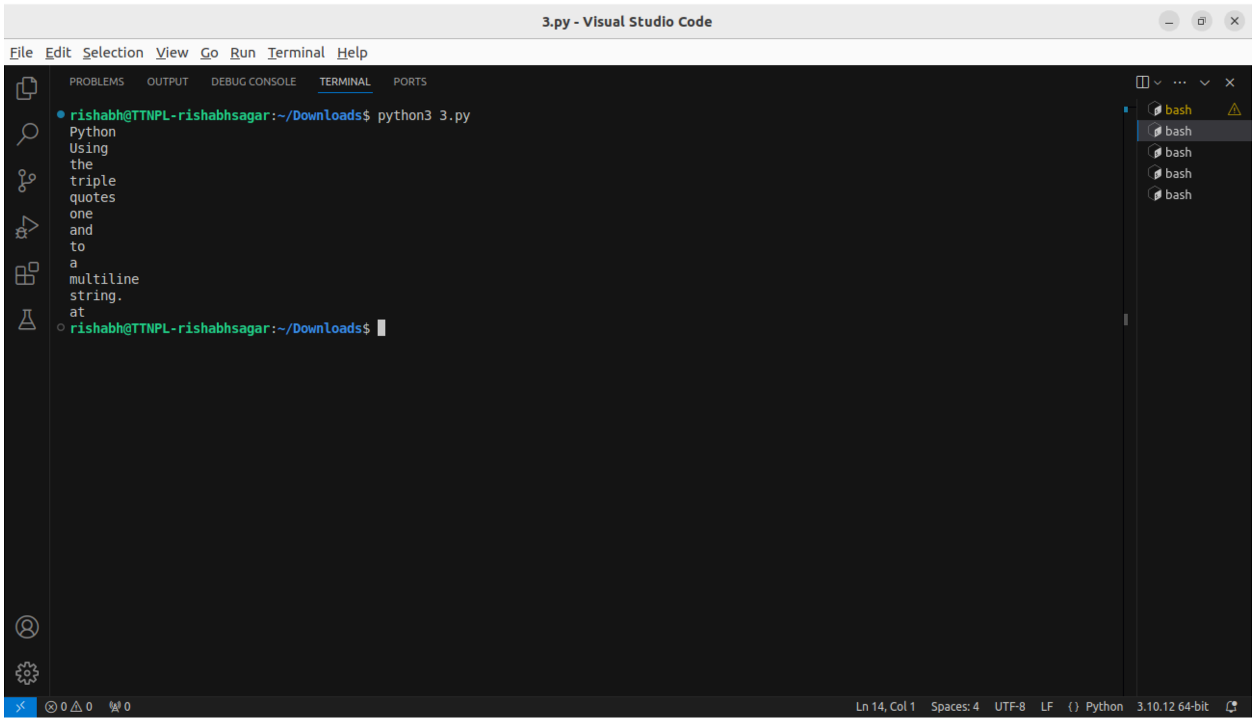
Q3)Programming: From a multi-words and multi-line string, prepare a filter for the list of words which have multiple occurrences in the string. Example Input: astring = """Python Multiline String Using Triple-Quotes Using the triple quotes style is one of the easiest and most common ways to split a large string into a multiline Python string. Triple quotes (''' or \""") can be used to create 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 start and one at the end, to define it.""" output: Python Using the triple quotes one and to a multiline string.

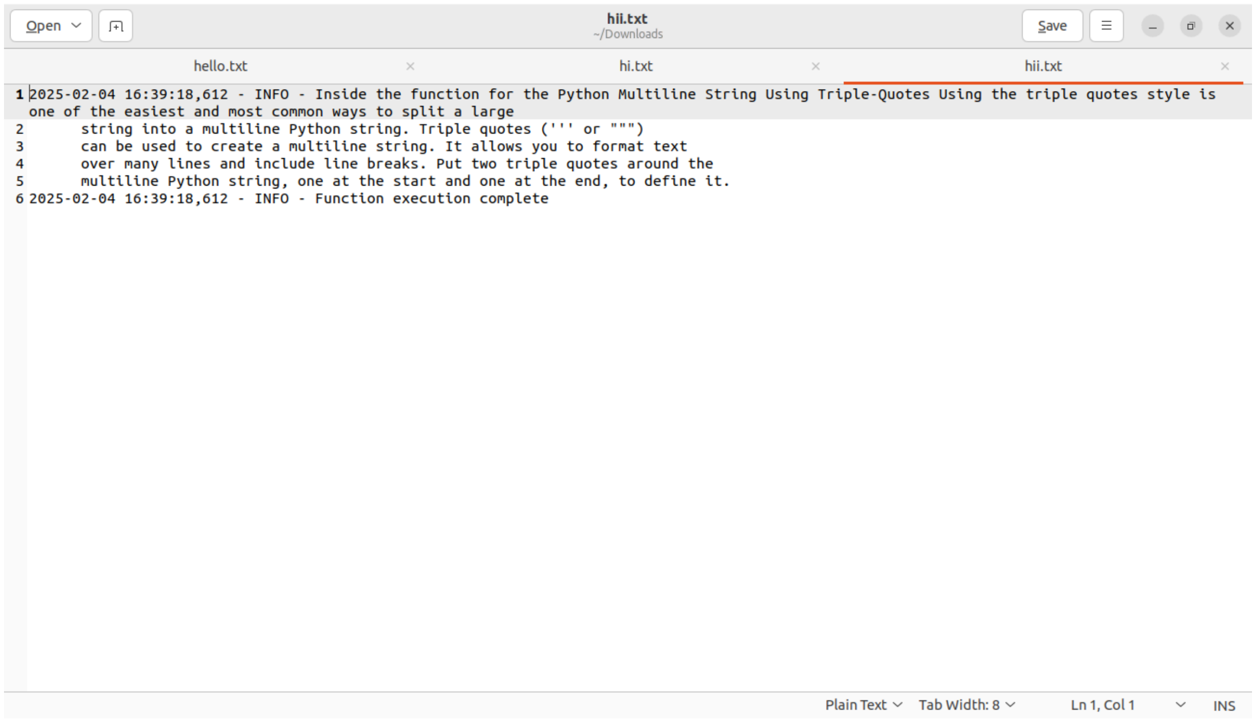
Ans.



```python
import logging
logging.basicConfig(filename="hii.txt", format="%(asctime)s - %(levelname)s - %(message)s",level=logging.INFO)
dic = {}
def count(stri):
    """Count function count the frequency of the word in the string and print only those word with frequency more than 1"""
    logging.info(f'Inside the function for the {stri}')
    sp = stri.split()
    for word in sp:
        if word in dic:
            dic[word] += 1
        else:
            dic[word] = 1
    logging.info("Function execution complete")
    ans = ([word for word, count in dic.items() if count > 1]) # make the final ans by taking only those word
    # which has frequency more than 1

    for i in ans:
        print(i)

count('''Python Multiline String Using Triple-Quotes Using the triple quotes style is one of the easiest and most common way
      string into a multiline Python string. Triple quotes (\''' or \""")
      can be used to create 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 start and one at the end, to define it.''')
```


3.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Ln 14, Col 1 Spaces: 4 UTF-8 LF Python 3.10.12 64-bit

```
rishabh@TTNPL-rishabhsagar:~/Downloads$ python3 3.py
Python
Using
the
triple
quotes
one
and
to
a
multiline
string.
at
rishabh@TTNPL-rishabhsagar:~/Downloads$
```

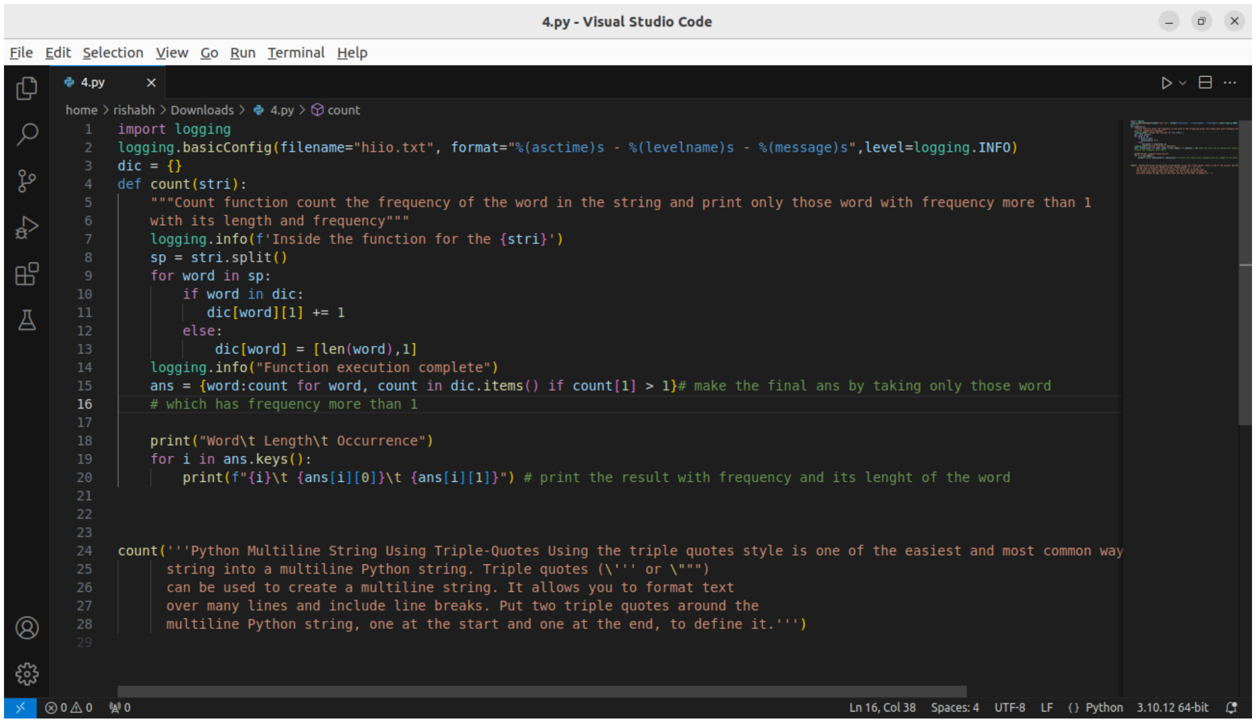


```
2025-02-04 16:39:18,612 - INFO - Inside the function for the Python Multiline String Using Triple-Quotes Using the triple quotes style is
one of the easiest and most common ways to split a large
       string into a multiline Python string. Triple quotes (''' or """)
       can be used to create 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 start and one at the end, to define it.
2025-02-04 16:39:18,612 - INFO - Function execution complete
```

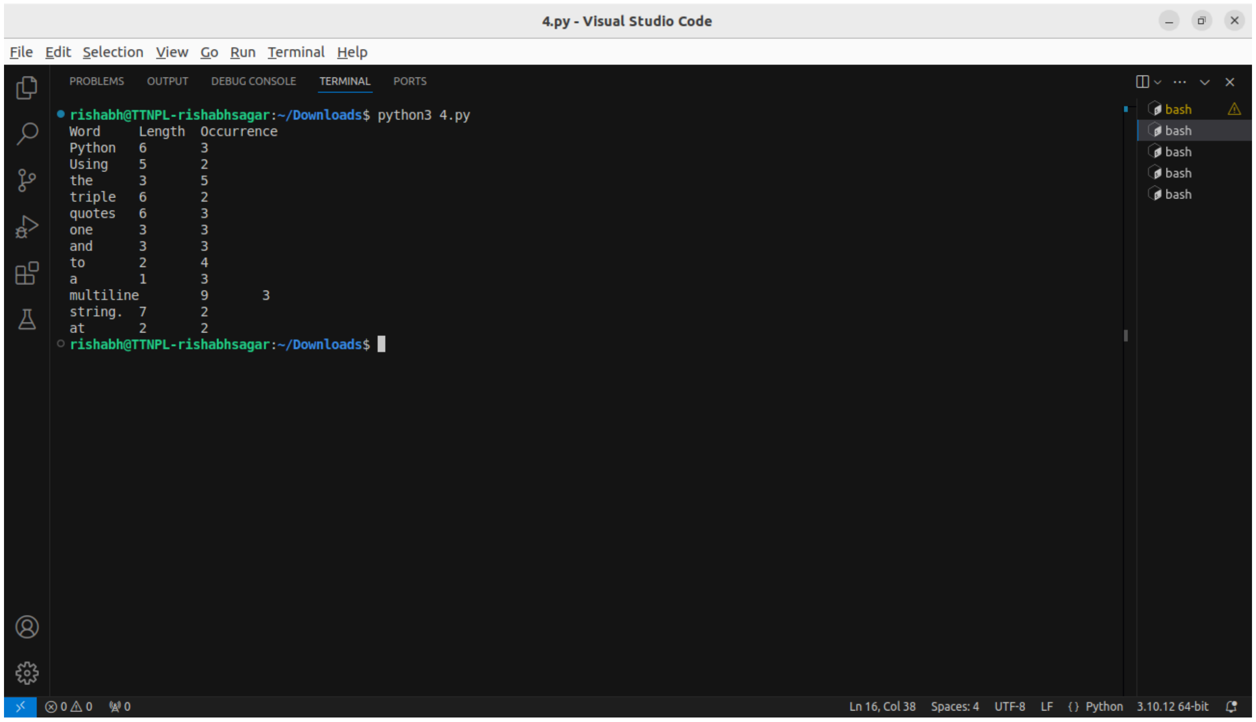
Q4)Programing: From a multi-words and multi-line string, display list of words and word's length with occurrence more than 1 in sorted order Example Input: astring = """Python Multiline String Using Triple-Quotes Using the triple quotes style is one of the easiest and most common ways to split a large string into a multiline Python string. Triple quotes (''' or \""") can be used to create 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 start and one at the end, to define it.""" 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

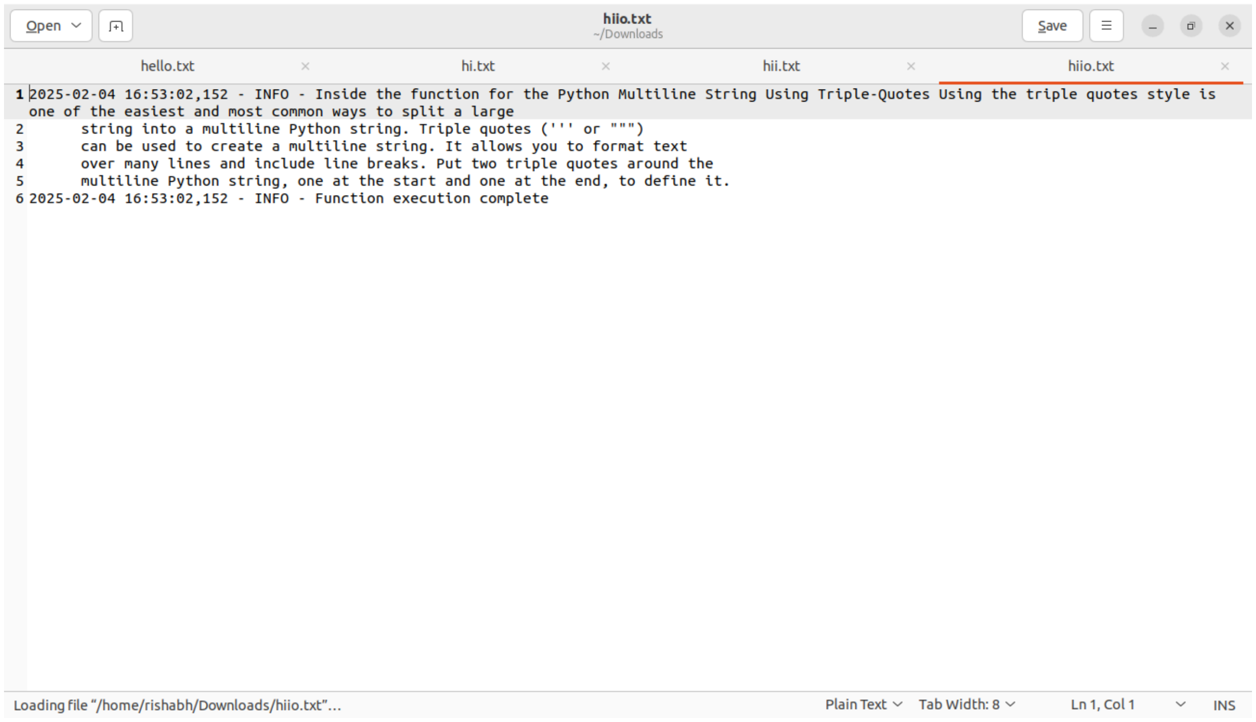
Ans.


4.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
4.py
home > rishabh > Downloads > 4.py > count
Ln 16, Col 38 Spaces: 4 UTF-8 LF {} Python 3.10.12 64-bit

```python
import logging
logging.basicConfig(filename="hiio.txt", format="%(asctime)s - %(levelname)s - %(message)s",level=logging.INFO)
dic = {}
def count(stri):
    """Count function count the frequency of the word in the string and print only those word with frequency more than 1
    with its length and frequency"""
    logging.info(f'Inside the function for the {stri}')
    sp = stri.split()
    for word in sp:
        if word in dic:
           dic[word][1] += 1
        else:
            dic[word] = [len(word),1]
    logging.info("Function execution complete")
    ans = {word:count for word, count in dic.items() if count[1] > 1}# make the final ans by taking only those word
    # which has frequency more than 1

    print("Word\t Length\t Occurrence")
    for i in ans.keys():
        print(f"{i}\t {ans[i][0]}\t {ans[i][1]}") # print the result with frequency and its lenght of the word



count('''Python Multiline String Using Triple-Quotes Using the triple quotes style is one of the easiest and most common way
      string into a multiline Python string. Triple quotes (\''' or \""")
      can be used to create 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 start and one at the end, to define it.''')
```


4.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
bash
bash
bash
bash
Ln 16, Col 38 Spaces: 4 UTF-8 LF {} Python 3.10.12 64-bit

```
rishabh@TTNPL-rishabhsagar:~/Downloads$ python3 4.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
rishabh@TTNPL-rishabhsagar:~/Downloads$
```



```
2025-02-04 16:53:02,152 - INFO - Inside the function for the Python Multiline String Using Triple-Quotes 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 be used to create 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 start and one at the end, to define it.
2025-02-04 16:53:02,152 - INFO - Function execution complete
```

Loading file "/home/rishabh/Downloads/hiio.txt"…

Plain Text   Tab Width: 8   Ln 1, Col 1   INS