Page No.
Date

Assignment 2

Class - SE IV
Batch - F4

ROLL NO - 21430
DOS - 26/9/2020

Problem Statement :-
Write a python program to compute following string operation.
a. To display word with the longest length.
b. To determine the frequency of concurence of particular character in the string.
c. To check whether given string is pallindrome or not
d. TO display the 1st index of a substring.
e. To count occurence of each word in the string.

Learning objective :-
1. To learn to write simple python program & execute it.
2. To learn implementation of string data structure.
3. To co learn & implement class and object.

Learning Outcomes :-
1. Will be able to write program in python & execute it.
2. Will be able to implement string data structure.
3. Will be able to implement class and object.

S/W & H/W requirement :-
1. Pycharm IDE Community Version 2020.
2. Windows 10 64 bit.
3. 8 GB Ram

Page No.
Date

Theory :-
String data structure

In python, strings are arrays of bytes representing unicode character. A string is a collection of one or more character put in single quote, double quote. In python their is a no char data type, a character is string of length one.

Syntax :-
string_name = 'string'
string_name = "string"
string_name = '''string'''

E.g. str = "Hello"

class and object :-

A class is an building block that leads to object oriented programming.

Syntax :
class class_name :
    #Statements

E.g. class Myclass :
    def __init__(self) :
        self.name = " "
    def show(self) :
        print(self.name)

Object is simply a variable of that class.

* Algorithm/ Psuedocode:-

* ADT for class string

```
Class string()
        str = " "   #empty string.
        longest()   #gave longest word in string
        frequency()  #gives freq of word.
        pallindrome()  # check string is pallindrome or not

        substring()   #returns 1st index of substring.
        occurence()   #return count of word in string
        read()        #read empty string
```

* Pseudocode for main()

```
START
string s
 s.read()
Read a
if a == 1
    s.longest()
if a == 2
    s.frequency()
if a == 3
    s.pallindrome()
if a == 4
    s.substring()
if a == 5
    s.occurence()
```

* Psudocode for longest()

```
START
longest (str)
    Ans = " "
    for each word in str
        if ( word.length() > ans.length() )
            ans = word
    END for
    return ans
END
```

* Psuedocode for frequency()

```
START
declare ans=0
for each character in string
    if ( character == req.character )
        ans : ans + 1
END for
return ans
END
```

* Psuedocode for Pallindrome()

```
START
n = string.length()
for i from 0 to n-1
    if ( string[i] not equal to string [n-i-1])
        Return not pallindrome
END for
return pallindrome
END
```

Psuedocode for Occurence()

```
START
list = list of words in string
for each element in list
    if count of word is present:
        count = count + 1
    else
        caunt = 0
END for
Return count
END.
```

* Complexity

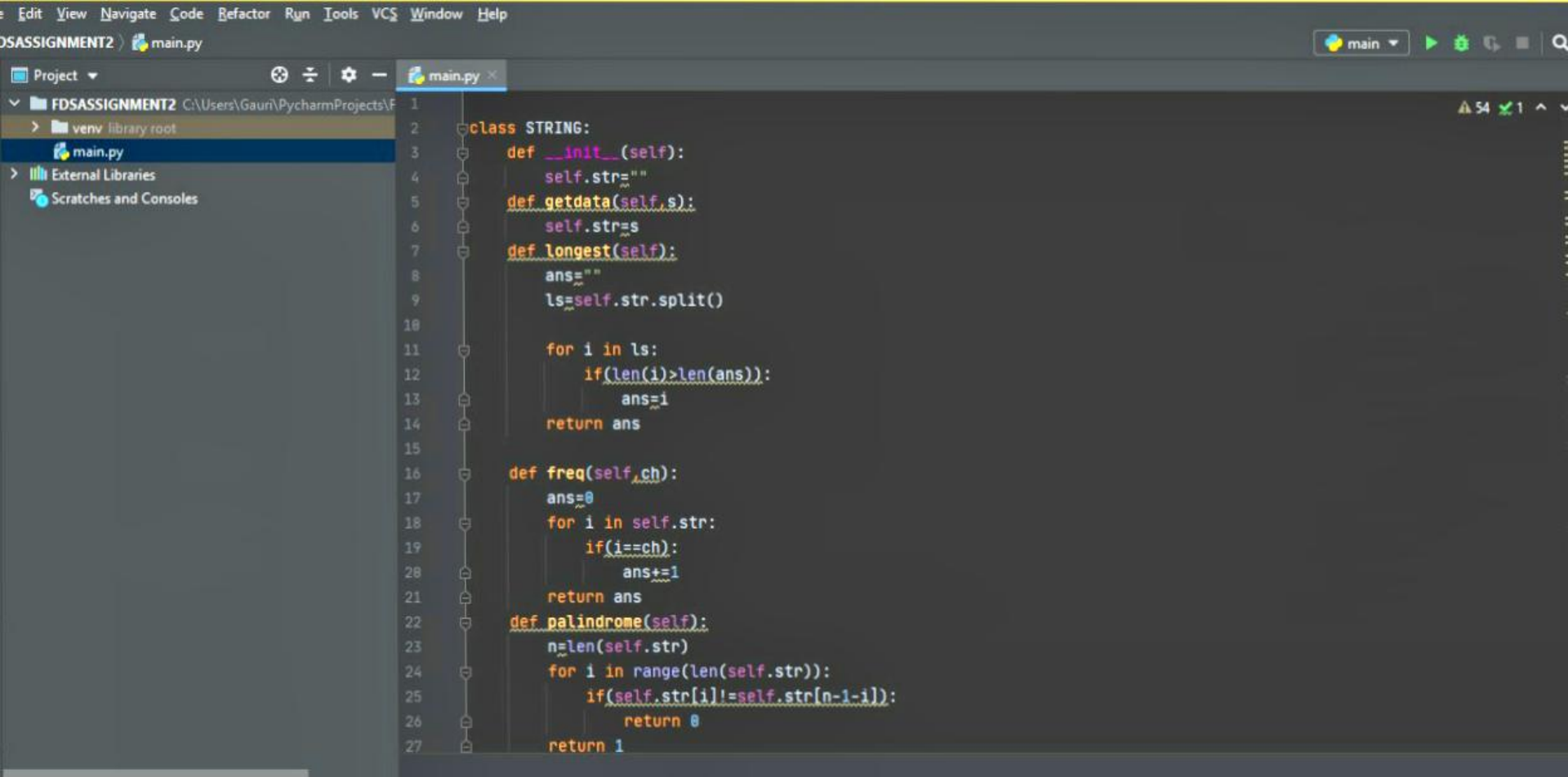
i. Occurence of word in string - O(n)
ii. pallindrome - O(n)
iii. Substring - O(n)
iv. longest word - O(n * len(str))

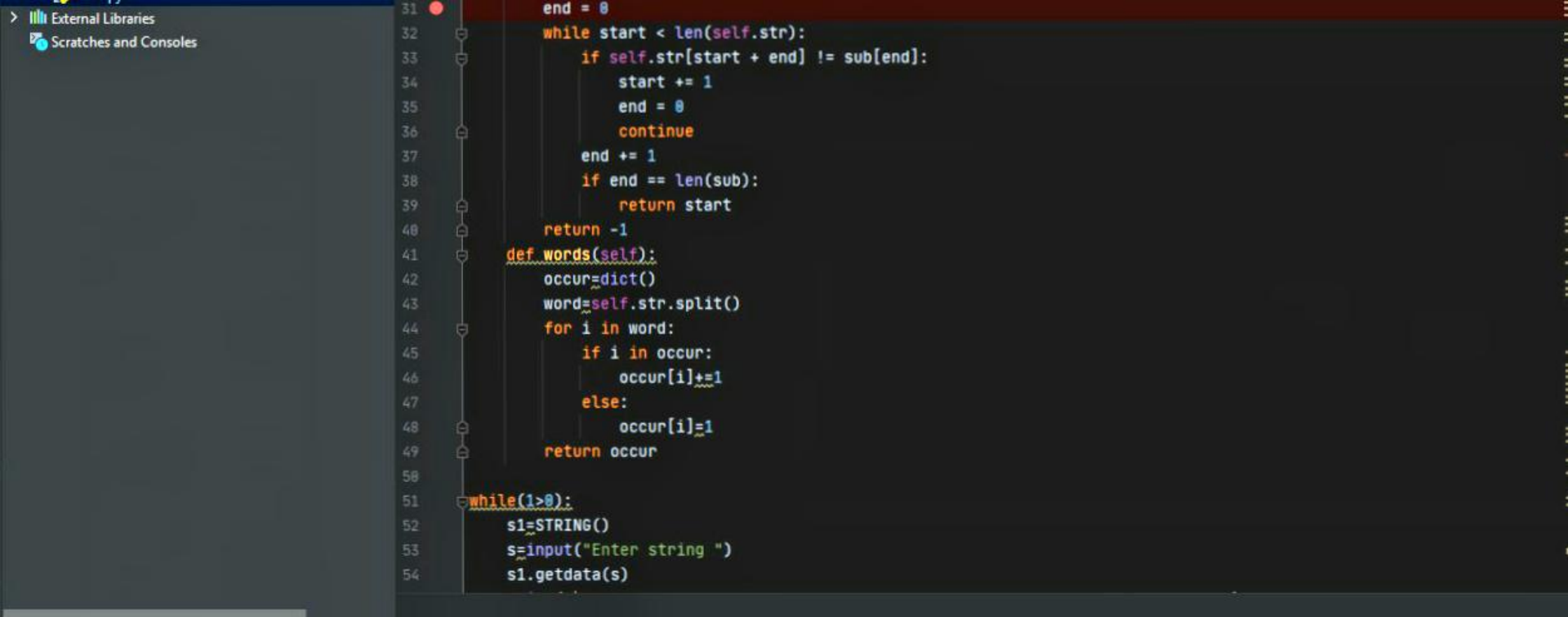
Page No.
Date

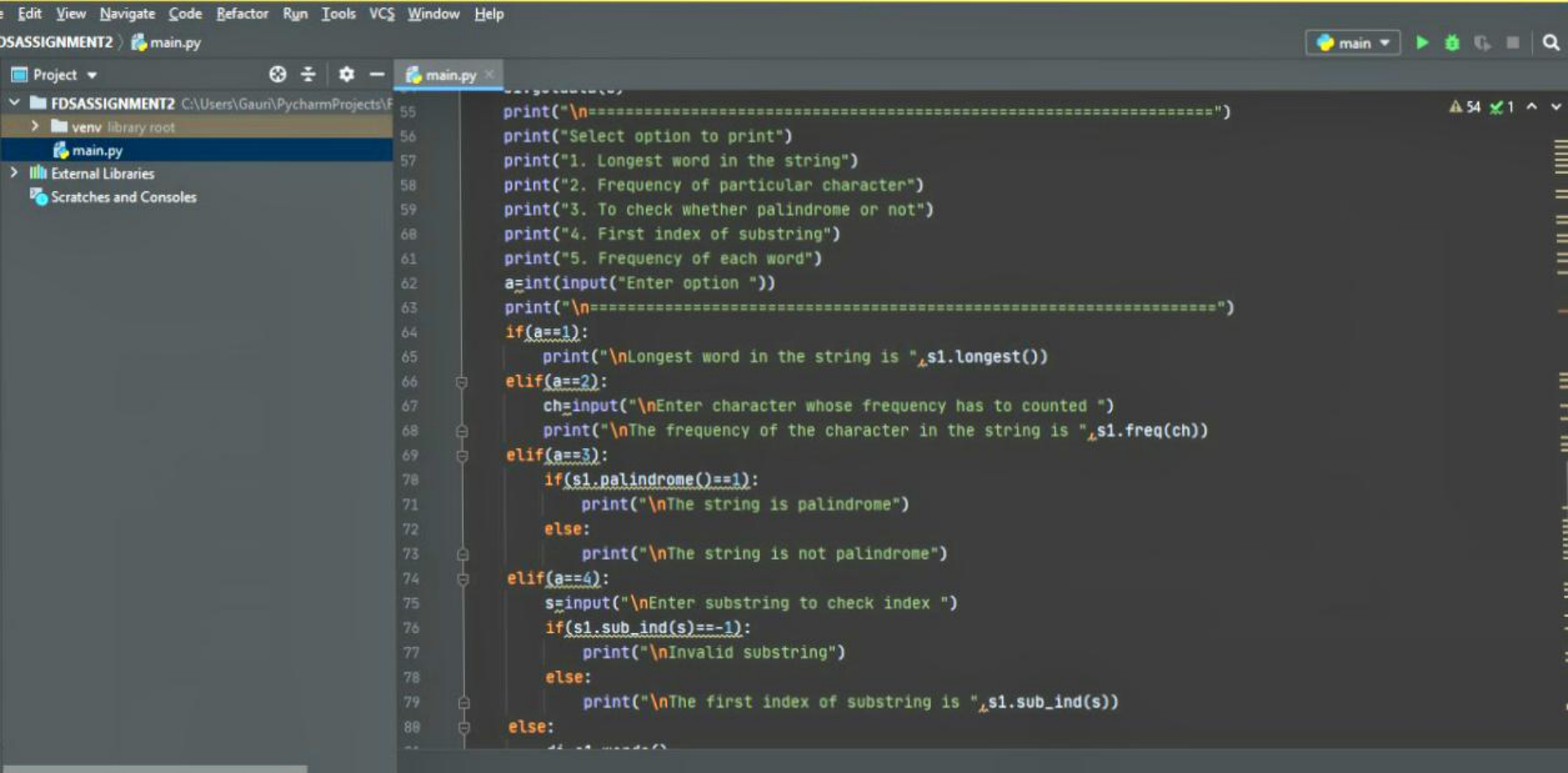
Test Cases:-

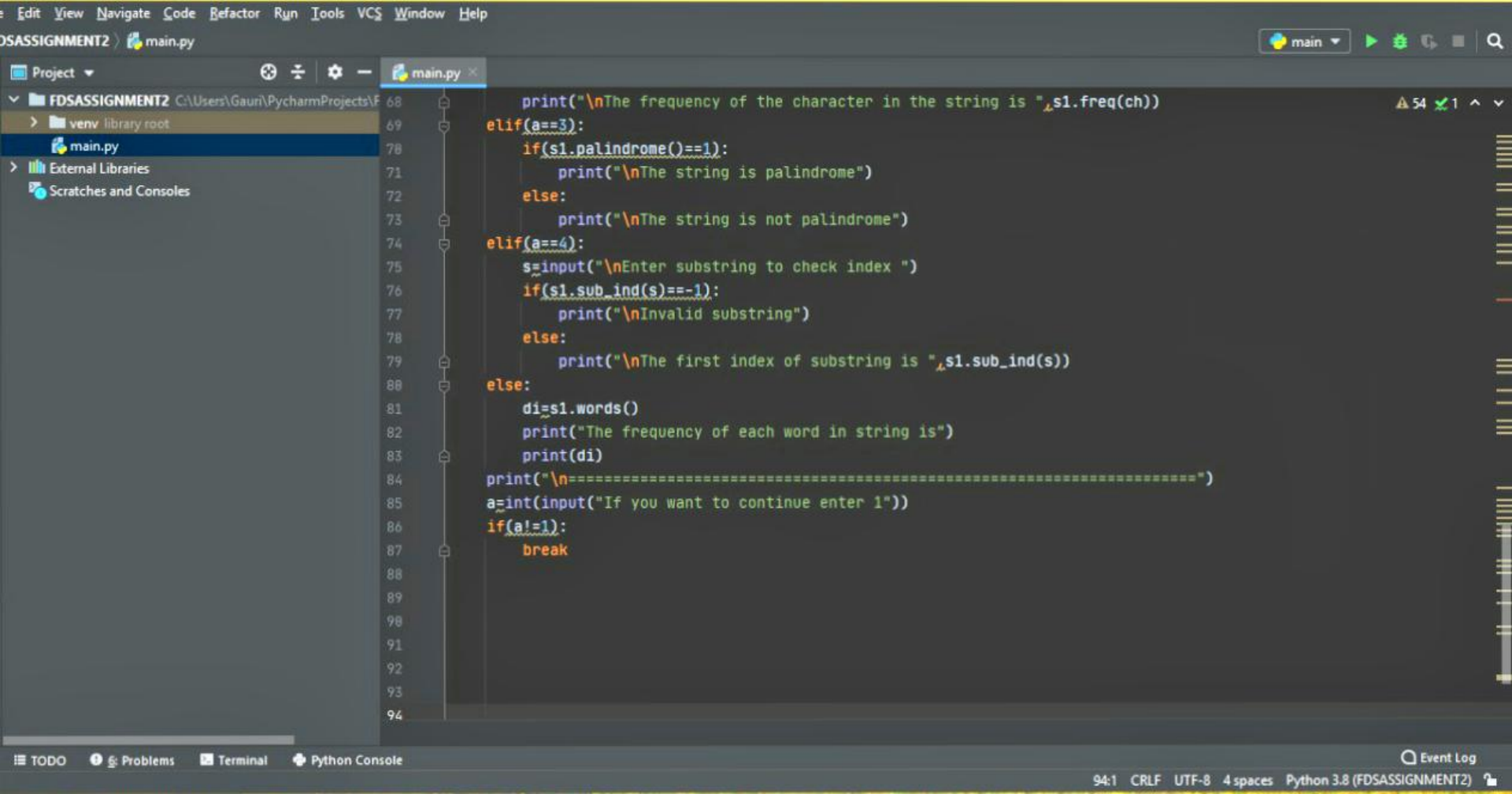
NO.	Description	Input	Expected O/P	Actual O/P	Result
1.	Menu:- 1. longest 2. Freq 3. Palindrome 4. Substring 5. occurence	str= Hello Worlds.	long is Worlds	long is Worlds	Pass.
2.	Menu:- 1. longest 2. Freq 3. Palindrome 4. Substring 5. occurence	str= Hello	Not Palim drome	Not Palli ndro me	Pass.

* Conclusion
He learnt about string data structure, class, object & implementation in python.


Edit View Navigate Code Refactor Run Tools VCS Window Help
DSASSIGNMENT2 main.py
main
Project
main.py
FDSASSIGNMENT2 C:\Users\Gauri\PycharmProjects\F
venv library root
main.py
External Libraries
Scratches and Consoles
class STRING:
def __init__(self):
self.str=""
def getdata(self,s):
self.str=s
def longest(self):
ans=""
ls=self.str.split()
for i in ls:
if(len(i)>len(ans)):
ans=i
return ans
def freq(self,ch):
ans=0
for i in self.str:
if(i==ch):
ans+=1
return ans
def palindrome(self):
n=len(self.str)
for i in range(len(self.str)):
if(self.str[i]!=self.str[n-1-i]):
return 0
return 1


External Libraries
Scratches and Consoles
end = 0
while start < len(self.str):
if self.str[start + end] != sub[end]:
start += 1
end = 0
continue
end += 1
if end == len(sub):
return start
return -1
def words(self):
occur=dict()
word=self.str.split()
for i in word:
if i in occur:
occur[i]+=1
else:
occur[i]=1
return occur
while(1>0):
s1=STRING()
s=input("Enter string ")
s1.getdata(s)


Edit View Navigate Code Refactor Run Tools VCS Window Help
DSASSIGNMENT2 main.py
main
Project
main.py
FDSASSIGNMENT2 C:\Users\Gauri\PycharmProjects\F
venv library root
main.py
External Libraries
Scratches and Consoles
print("\n==========")
print("Select option to print")
print("1. Longest word in the string")
print("2. Frequency of particular character")
print("3. To check whether palindrome or not")
print("4. First index of substring")
print("5. Frequency of each word")
a=int(input("Enter option "))
print("\n==========")
if(a==1):
print("\nLongest word in the string is ",s1.longest())
elif(a==2):
ch=input("\nEnter character whose frequency has to counted ")
print("\nThe frequency of the character in the string is ",s1.freq(ch))
elif(a==3):
if(s1.palindrome()==1):
print("\nThe string is palindrome")
else:
print("\nThe string is not palindrome")
elif(a==4):
s=input("\nEnter substring to check index ")
if(s1.sub_ind(s)==-1):
print("\nInvalid substring")
else:
print("\nThe first index of substring is ",s1.sub_ind(s))
else:


Edit View Navigate Code Refactor Run Tools VCS Window Help
DSASSIGNMENT2 main.py
main
Project
main.py
FDSASSIGNMENT2 C:\Users\Gauri\PycharmProjects\F
venv library root
main.py
External Libraries
Scratches and Consoles
print("\nThe frequency of the character in the string is ",s1.freq(ch))
elif(a==3):
if(s1.palindrome()==1):
print("\nThe string is palindrome")
else:
print("\nThe string is not palindrome")
elif(a==4):
s=input("\nEnter substring to check index ")
if(s1.sub_ind(s)==-1):
print("\nInvalid substring")
else:
print("\nThe first index of substring is ",s1.sub_ind(s))
else:
di=s1.words()
print("The frequency of each word in string is")
print(di)
print("\n==")
a=int(input("If you want to continue enter 1"))
if(a!=1):
break
54 1
TODO
6: Problems
Terminal
Python Console
Event Log
94:1 CRLF UTF-8 4 spaces Python 3.8 (FDSASSIGNMENT2)

```
C:\Users\Gauri\PycharmProjects\FDSASSIGNMENT2\venv\Scripts\python.exe C:/Users/Gauri/PycharmProjects/FDSASSIGNMENT2/main.py
Enter string Ganesh KAndepalli

======================================================================
Select option to print
1. Longest word in the string
2. Frequency of particular character
3. To check whether palindrome or not
4. First index of substring
5. Frequency of each word
Enter option 1

======================================================================

Longest word in the string is  KAndepalli

======================================================================
If you want to continue enter 1
Enter string Ganesh KAndepalli

======================================================================
Select option to print
1. Longest word in the string
2. Frequency of particular character
3. To check whether palindrome or not
4. First index of substring
5. Frequency of each word
```

```
4. First index of substring
5. Frequency of each word
Enter option 2

======================================================================

Enter character whose frequency has to counted a

The frequency of the character in the string is  2

======================================================================
If you want to continue enter 1 1
Enter string Ganesh KAndepalli

======================================================================
Select option to print
1. Longest word in the string
2. Frequency of particular character
3. To check whether palindrome or not
4. First index of substring
5. Frequency of each word
Enter option 3

======================================================================

The string is not palindrome
```

```
======================================================================
If you want to continue enter 1: 1
Enter string: Ganesh KAndepalli

======================================================================
Select option to print
1. Longest word in the string
2. Frequency of particular character
3. To check whether palindrome or not
4. First index of substring
5. Frequency of each word
Enter option: 4

======================================================================

Enter substring to check index: I

Invalid substring

======================================================================
If you want to continue enter 1: 1
Enter string: Ganesh KAndepalli

======================================================================
Select option to print
1. Longest word in the string
```

```
Enter substring to check index 1

Invalid substring

========================================================================
If you want to continue enter 11
Enter string Ganesh KAndepalli

========================================================================
Select option to print
1. Longest word in the string
2. Frequency of particular character
3. To check whether palindrome or not
4. First index of substring
5. Frequency of each word
Enter option 5

========================================================================
The frequency of each word in string is
{'Ganesh': 1, 'KAndepalli': 1}

========================================================================
If you want to continue enter 12

Process finished with exit code 0
```