ITW1

PYTHON ASSIGNMENT2

tarun.arora.cse20@itbhu.ac.in

Name: Tarun Arora

Roll No.: 20075092

Branch: CSE

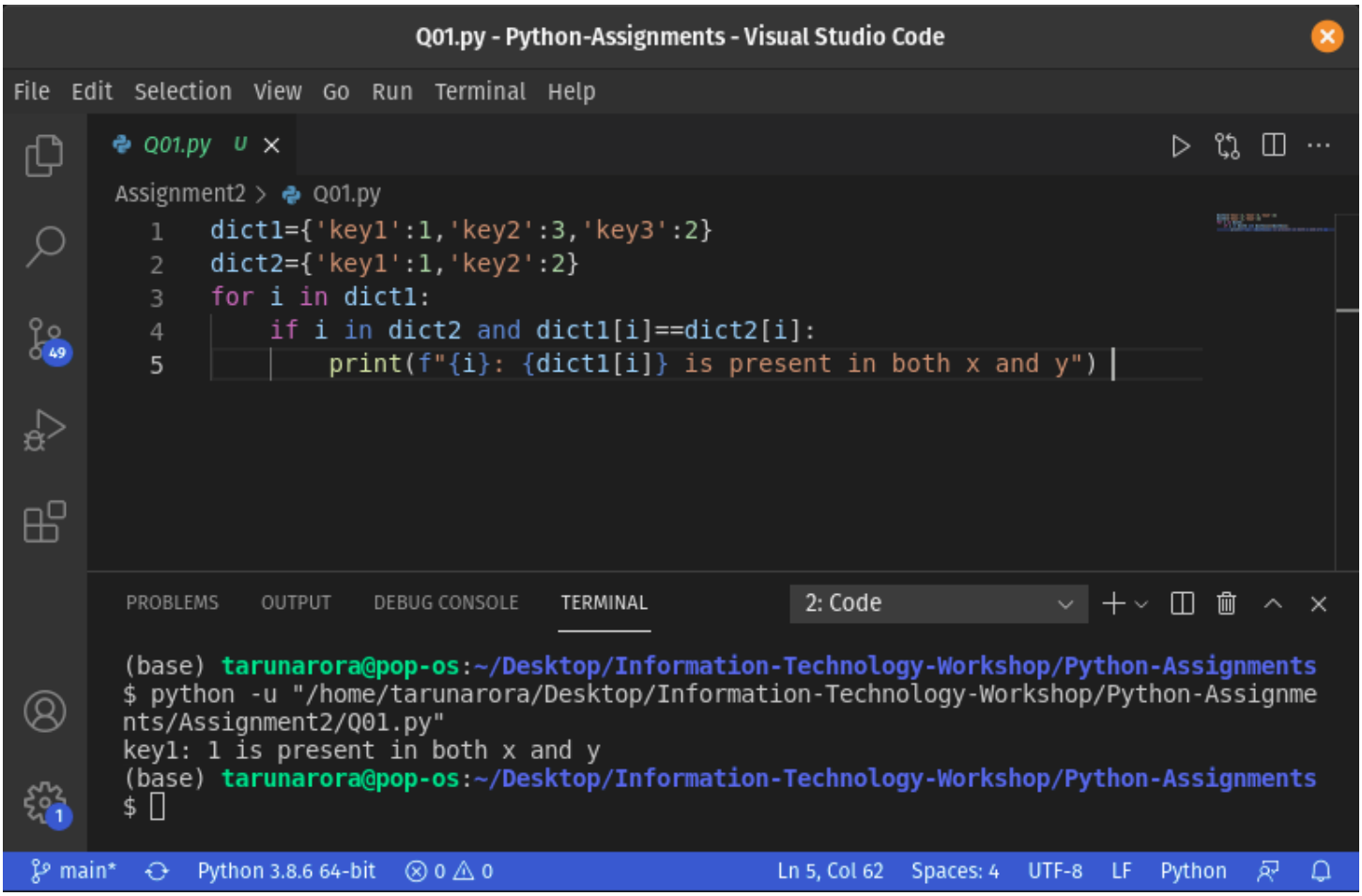
Date: July 20, 2021

1. Write a Python program to match key values in two dictionaries.

Sample dictionary: {'key1': 1, 'key2': 3, 'key3': 2}, {'key1': 1, 'key2': 2}

Expected output: key1: 1 is present in both x and y.

Solution: -

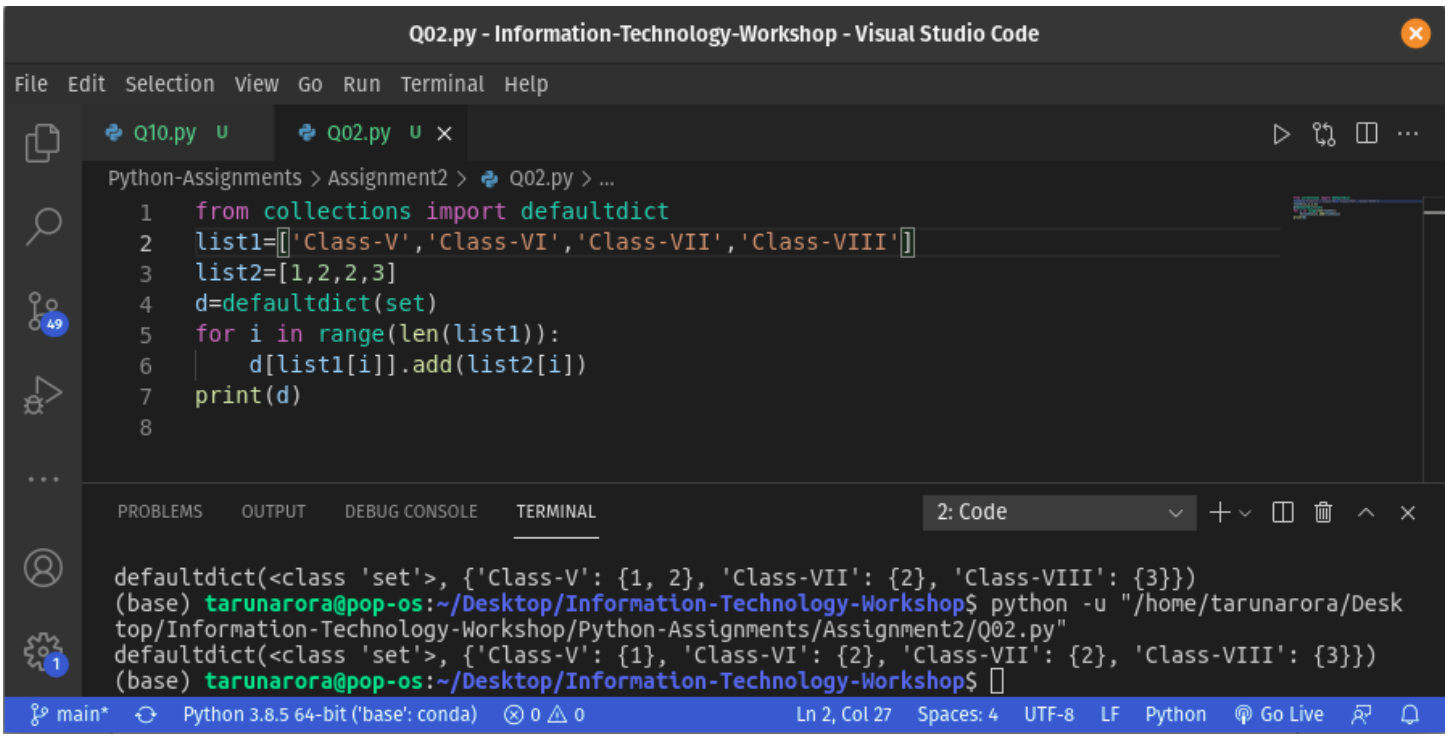


2. **Write a Python program to create a dictionary from two lists without losing duplicate values.**

Sample lists: ['Class-V', 'Class-VI', 'Class-VII', 'Class-VIII'], [1, 2, 2, 3]

Expected Output: defaultdict(<class 'set'>, {'Class-VII': {2}, 'Class-VI': {2}, 'Class-VIII': {3}, 'Class-V': {1}})

Solution: -



3. Write a Python program to replace dictionary values with their sum.

Example: Input: {'id' : 1, 'subject' : 'math', 'V' : 70, 'VI': 82},

{'id' : 2, 'subject' : 'math', 'V' : 73, 'VI' : 74},

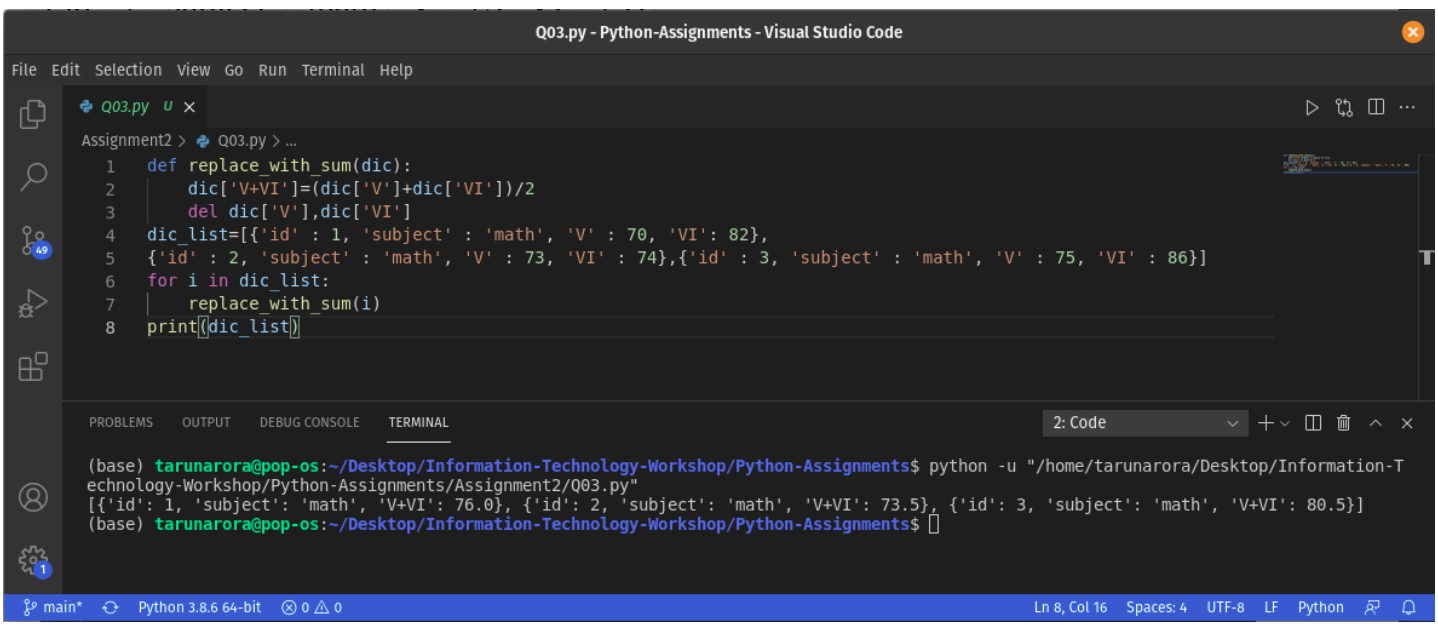
{'id' : 3, 'subject' : 'math', 'V' : 75, 'VI' : 86}

Output: [{'subject': 'math', 'id': 1, 'V+VI': 76.0},

{'subject': 'math', 'id': 2, 'V+VI': 73.5}

{'subject': 'math', 'id': 3, 'V+VI': 80.5}]

Solution: -

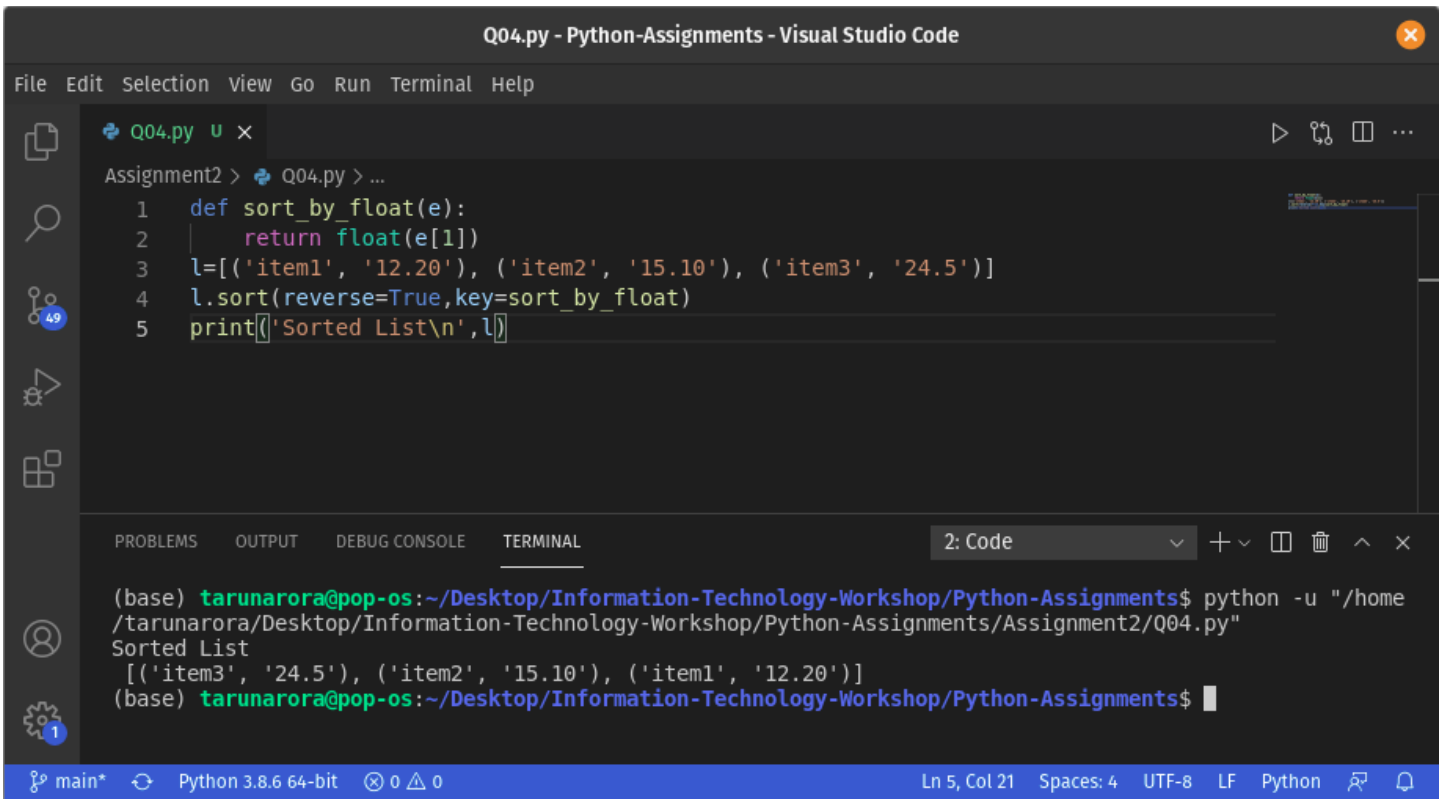


4. Write a Python program to sort a tuple by its float element.

Sample data: [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')]

Expected Output: [('item3', '24.5'), ('item2', '15.10'), ('item1', '12.20')]

Solution: -

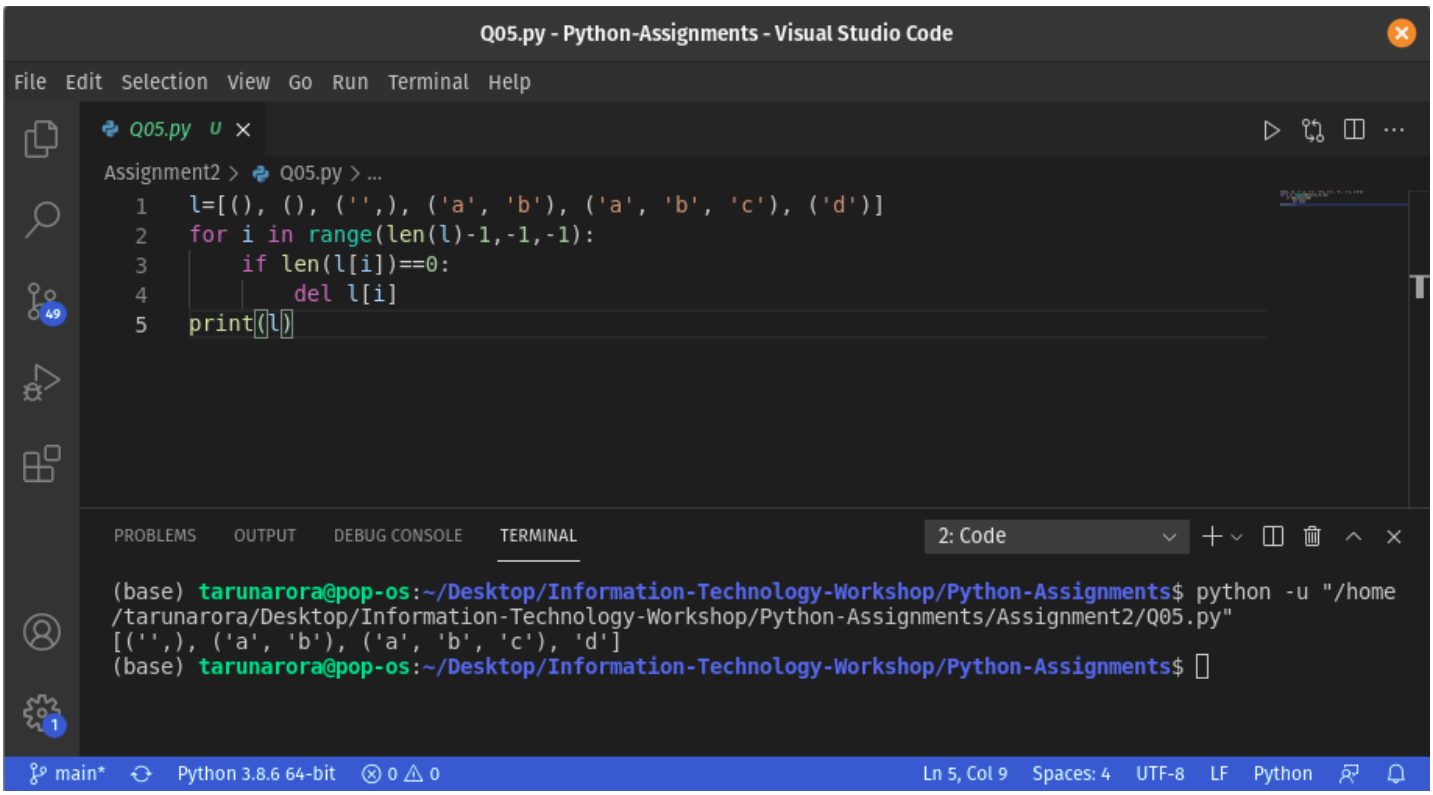


5. Write a Python program to remove an empty tuple(s) from a list of tuples.

Sample data: [(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')]

Expected output: [('',), ('a', 'b'), ('a', 'b', 'c'), 'd']

Solution: -

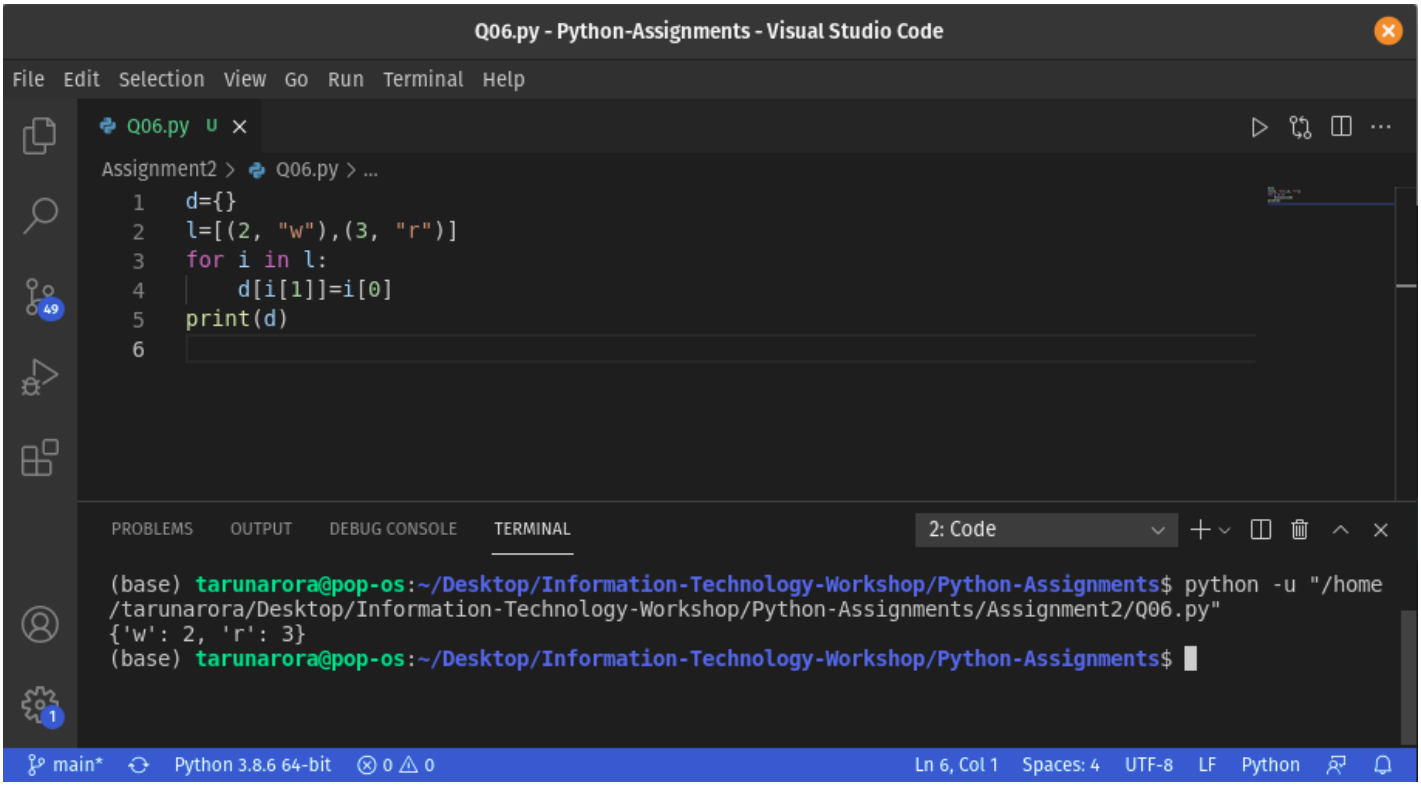


6. Write a Python program to convert a list of tuples into a dictionary.

Example: Input: ((2, "w"),(3, "r"))

Output: {'w': 2, 'r': 3}

Solution: -

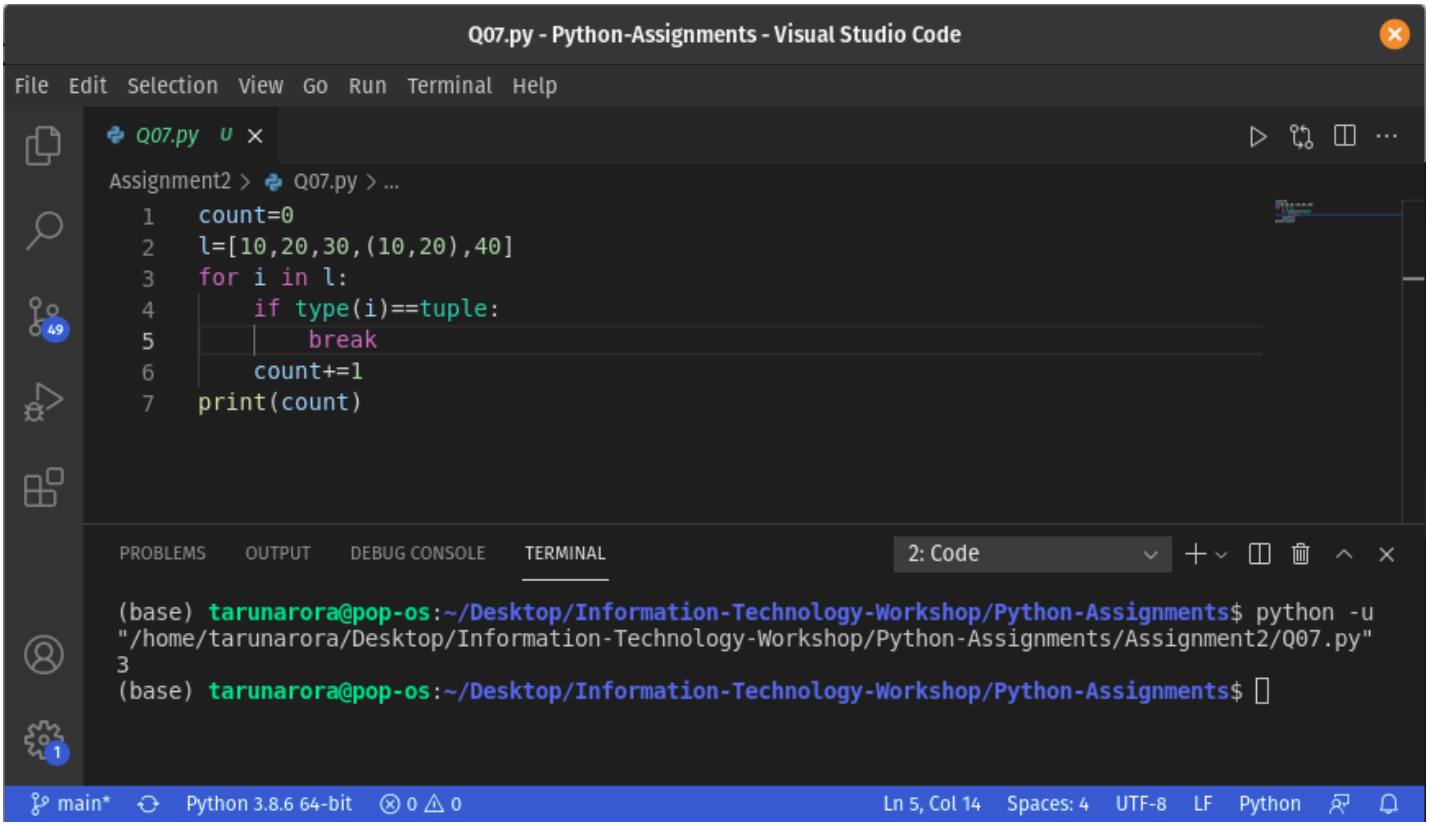


7. **Write a Python program to count the elements in a list until an element is a tuple.**

Example: Input: [10,20,30, (10,20),40]

Output: 3

Solution: -

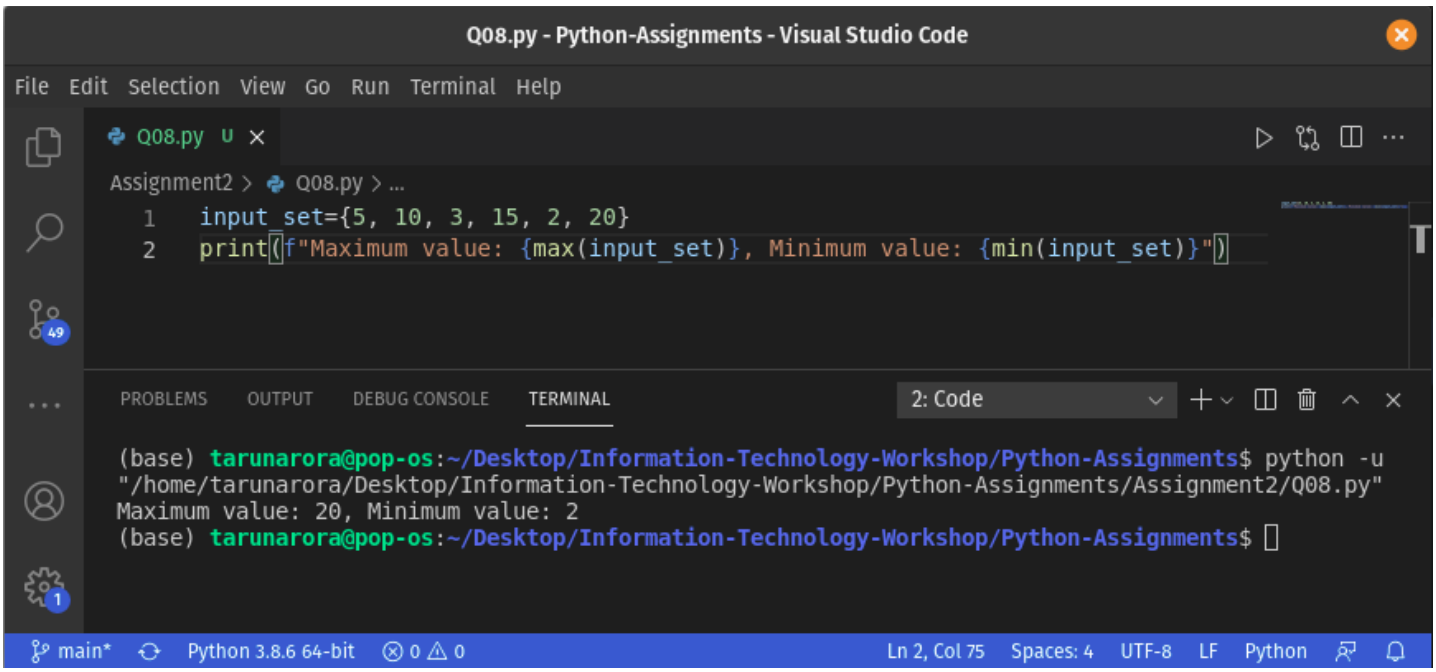


8. Write a Python program to find maximum and the minimum value in a set.

Example: Input: ([5, 10, 3, 15, 2, 20])

Output: Maximum value: 20, Minimum value: 2

Solution: -



9. **Write a Python program to create set difference, union, and intersection of sets.**

Example: Input: set(["green", "blue"]), set(["blue", "yellow"])

Output: Difference: {'blue'}, {'green'}, Union: {'yellow', 'green', 'blue'}, Intersection: {'blue'}

Solution: -

Q09.py - Python-Assignments - Visual Studio Code

File Edit Selection View Go Run Terminal Help

Q09.py U

Assignment2 > Q09.py > ...

```python
union=set([])
difference=set([])
intersection=set([])
set1= set(["green", "blue"])
set2= set(["blue", "yellow"])
for i in set1:
    if i in set2:
        intersection.add(i)
    else:
        difference.add(i)
    union.add(i)
for i in set2:
    union.add(i)
print(f"Difference: {difference}, Union: {union}, Intersection: {intersection}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Code

```
(base) tarunarora@pop-os:~/Desktop/Information-Technology-Workshop/Python-Assignments$ python -u "/home/tarunarora/Desktop/Information-Technology-Workshop/Python-Assignments/Assignment2/Q09.py"
Difference: {'green'}, Union: {'blue', 'yellow', 'green'}, Intersection: {'blue'}
(base) tarunarora@pop-os:~/Desktop/Information-Technology-Workshop/Python-Assignments$
```

main* Python 3.8.6 64-bit 0 0 Ln 12, Col 15 Spaces: 4 UTF-8 LF Python

10. Write a Python program to make a chain of function decorators (bold, italic, underline etc.).

Example: Input: hello world

Output: hello world

Solution: -

Q10.py - Information-Technology-Workshop - Visual Studio Code

File Edit Selection View Go Run Terminal Help

Q10.py U

Python-Assignments > Assignment2 > Q10.py > hello

```python
def makeBold(fn):
    def wrapped():
        return '\033[1m'+fn()+'\033[0m'
    return wrapped
def makeItalic(fn):
    def wrapped():
        return '\x1B[3m'+fn()+'\x1B[23m'
    return wrapped
def doUnderline(fn):
    def wrapped():
        return '\033[4m'+fn()+'\033[0m'
    return wrapped
@makeBold
@makeItalic
@doUnderline
def hello():
    s=input('Enter Text: ')
    return s
print(hello())
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Code

```
(base) tarunarora@pop-os:~/Desktop/Information-Technology-Workshop$ python -u "/home/taru
narora/Desktop/Information-Technology-Workshop/Python-Assignments/Assignment2/Q10.py"
Enter Text: hello world
hello world
(base) tarunarora@pop-os:~/Desktop/Information-Technology-Workshop$
```

main* Python 3.8.5 64-bit ('base': conda) 0 0 Ln 18, Col 13 Spaces: 4 UTF-8 LF Python Go Live

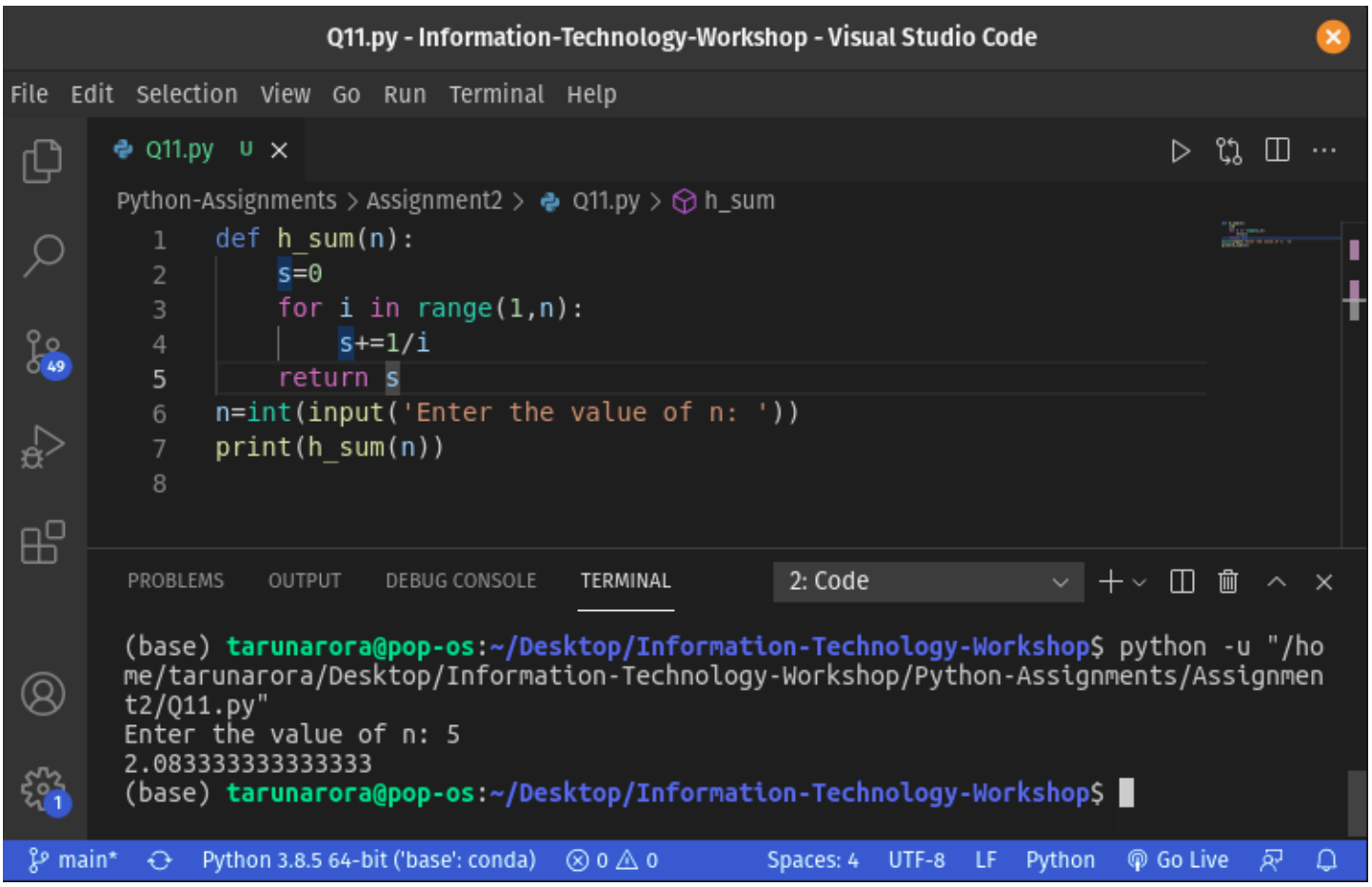
11. Write a Python program to calculate the harmonic sum of n-1.

Note: The harmonic sum is the sum of reciprocals of the positive integers.

Example :

$$1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\cdots$$

Solution: -

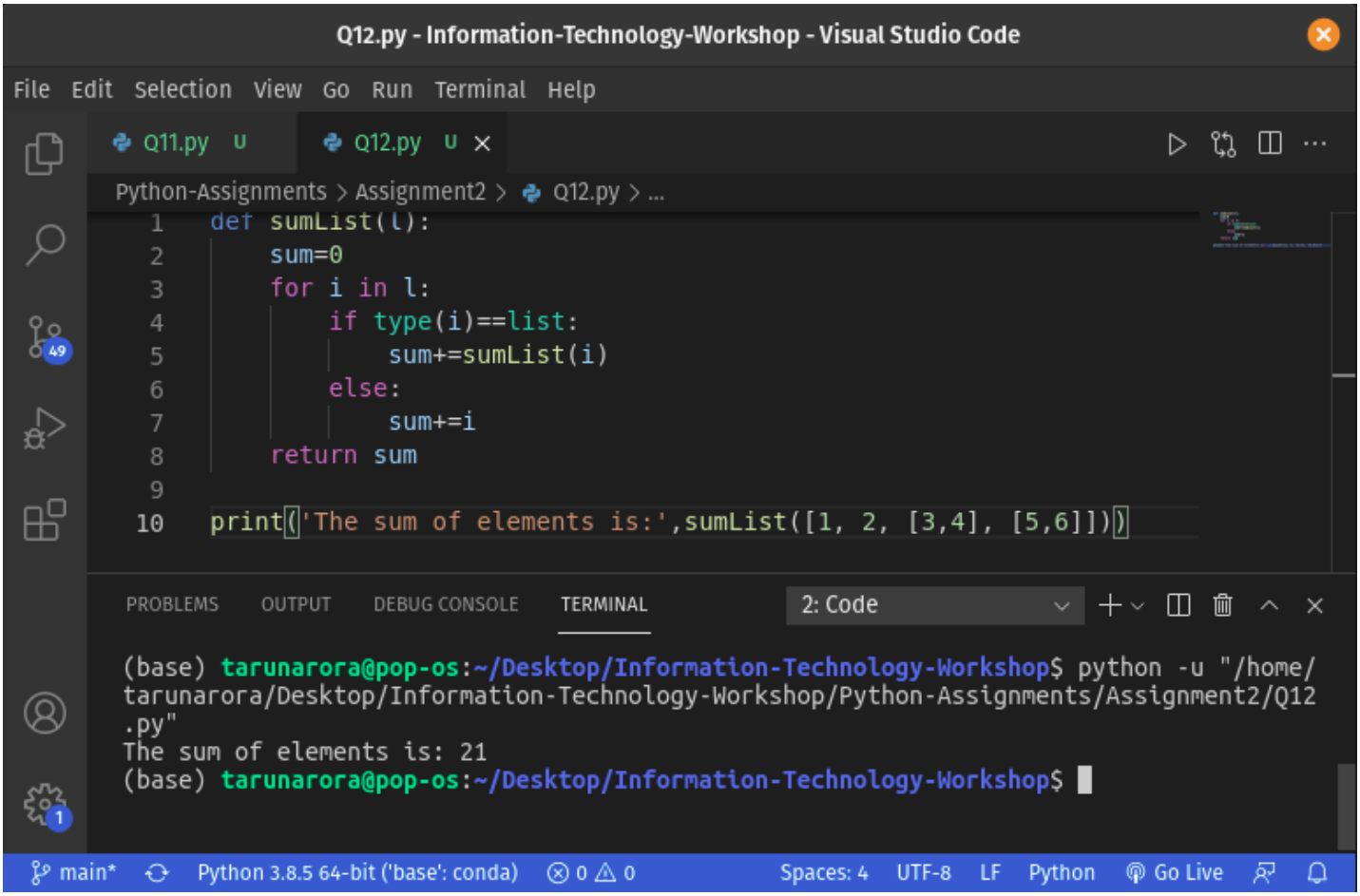


12. **Write a Python program of recursion list sum.**

Test Data: [1, 2, [3,4], [5,6]]

Expected Result: 21

Solution: -



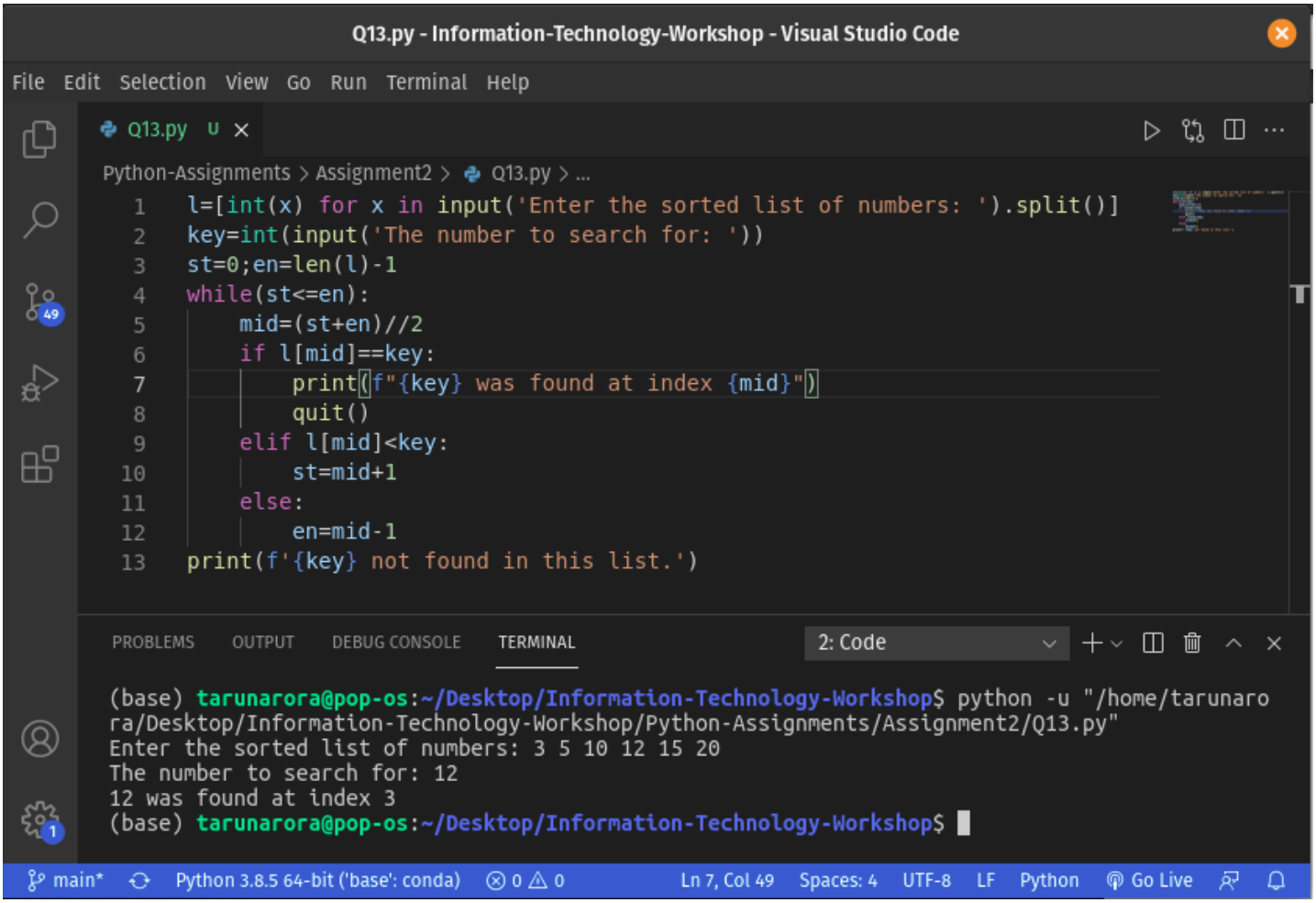
13. Write a Python program for binary search.

Example: Enter the sorted list of numbers: 3 5 10 12 15 20

The number to search for: 12

12 was found at index 3

Solution: -



14. Write a Python program to sort a list of elements using the bubble sort algorithm.

Example: Sample Data: [14, 46, 43, 27, 57, 41, 45, 21, 70]

Expected Result: [14, 21, 27, 41, 43, 45, 46, 57, 70]

Solution: -

Q14.py - Information-Technology-Workshop - Visual Studio Code

File Edit Selection View Go Run Terminal Help

Q14.py U

Python-Assignments > Assignment2 > Q14.py > ...

```python
l=[int(x) for x in input('Enter a list of integers you want to sort: ').split()]
def bubbleSort(l):
    for i in range(len(l)-1):
        for j in range(len(l)-i-1):
            if l[j+1]<l[j]:
                l[j+1],l[j]=l[j],l[j+1]
print('The given list to be sorted is:',l)
bubbleSort(l)
print('The sorted list is:',l)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Code

```
(base) tarunarora@pop-os:~/Desktop/Information-Technolo
gy-Workshop$ python -u "/home/tarunarora/Desktop/Information-Technology-W
orkshop/Python-Assignments/Assignment2/Q14.py"
Enter a list of integers you want to sort: 6 52 14 23 89 11 0 -1
The given list to be sorted is: [6, 52, 14, 23, 89, 11, 0, -1]
The sorted list is: [-1, 0, 6, 11, 14, 23, 52, 89]
(base) tarunarora@pop-os:~/Desktop/Information-Technology-Workshop$
```

main* Python 3.8.5 64-bit ('base': conda) 0 0 Ln 9, Col 31 Spaces: 4 UTF-8 LF Python Go Live

15. Write a Python program to sort a list of elements using the selection sort algorithm.

Example: Sample Data: [14, 46, 43, 27, 57, 41, 45, 21, 70]

Expected Result: [14, 21, 27, 41, 43, 45, 46, 57, 70]

Solution: -

Q15.py - Information-Technology-Workshop - Visual Studio Code

File Edit Selection View Go Run Terminal Help

Q15.py U

Python-Assignments > Assignment2 > Q15.py > ...

```python
l=[int(x) for x in input('Enter a list of integers you want to sort: ').split()]
def selectionSort(l):
    for top in range(len(l)-1):
        k=top
        for i in range(top+1,len(l)):
            if l[i]<l[k]:
                k=i
        l[top],l[k]=l[k],l[top]
print('The given list to be sorted is:',l)
selectionSort(l)
print('The sorted list is:',l)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Code

```
(base) tarunarora@pop-os:~/Desktop/Information-Technology-Workshop$ python -u "/home/tarunarora/Desktop/Information-Technology-Workshop/Python-Assignments/Assignment2/Q15.py"
Enter a list of integers you want to sort: 6 52 14 23 89 11 0 -1
The given list to be sorted is: [6, 52, 14, 23, 89, 11, 0, -1]
The sorted list is: [-1, 0, 6, 11, 14, 23, 52, 89]
(base) tarunarora@pop-os:~/Desktop/Information-Technology-Workshop$
```

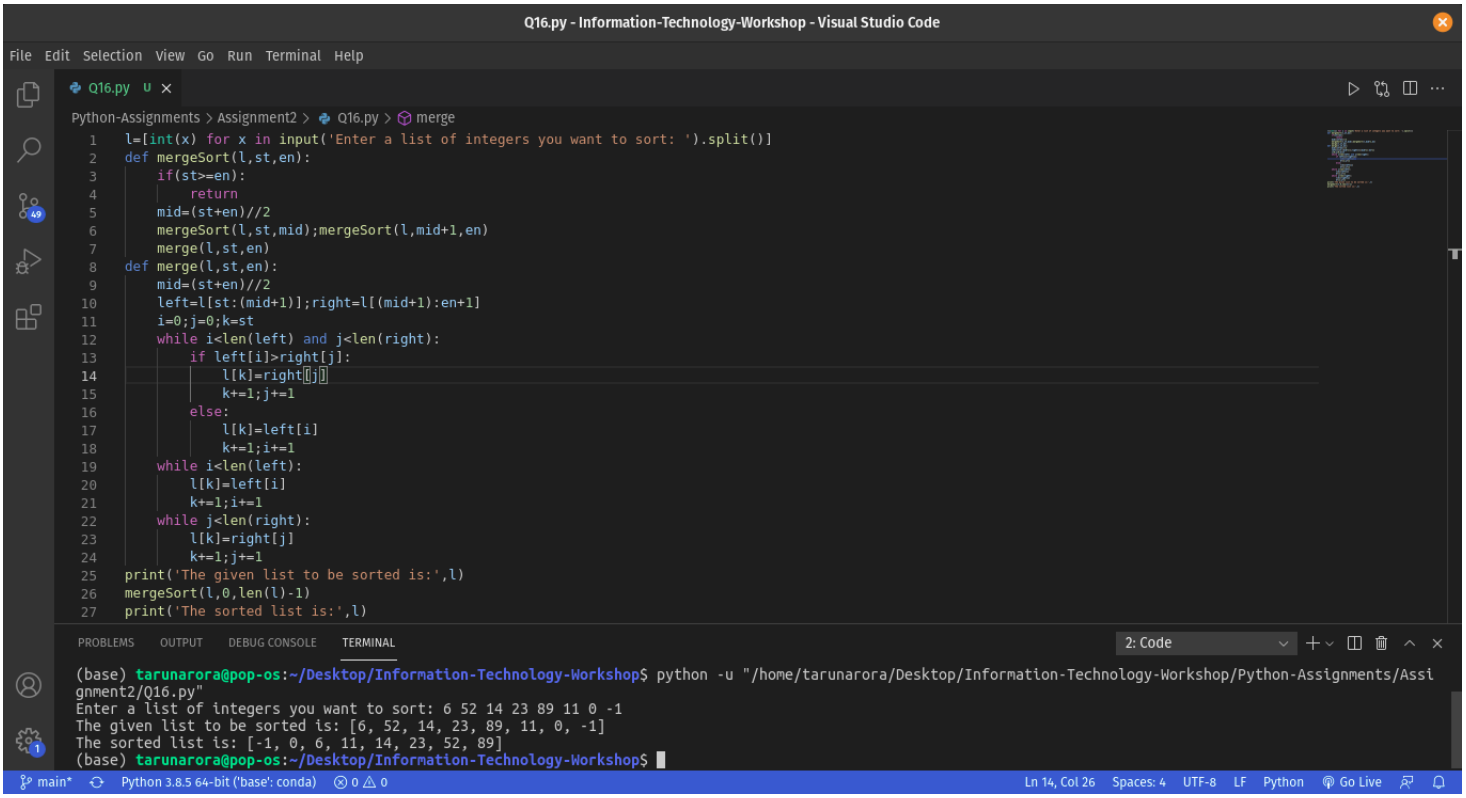
main* Python 3.8.5 64-bit ('base': conda) 0 0 Ln 9, Col 43 Spaces: 4 UTF-8 LF Python Go Live

16. Write a Python program to sort a list of elements using the merge sort algorithm.

Example: Split Sample Data: [14, 46, 43, 27, 57, 41, 45, 21, 70]

Merge and Sort(Expected Result): [14, 21, 27, 41, 43, 45, 46, 57, 70]

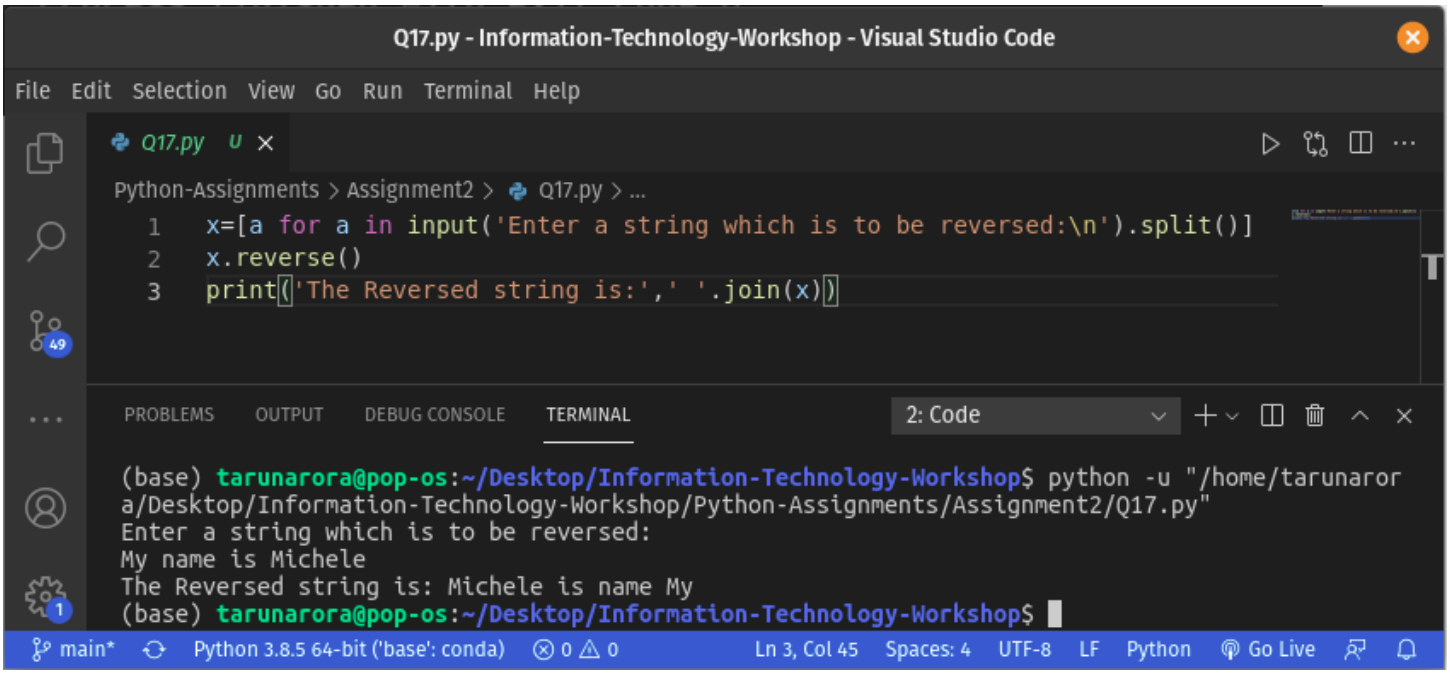
Solution: -



17. Write a Python program using functions that asks the user for a long string containing multiple words. Print back to the user the same string, except with the words in backwards order.

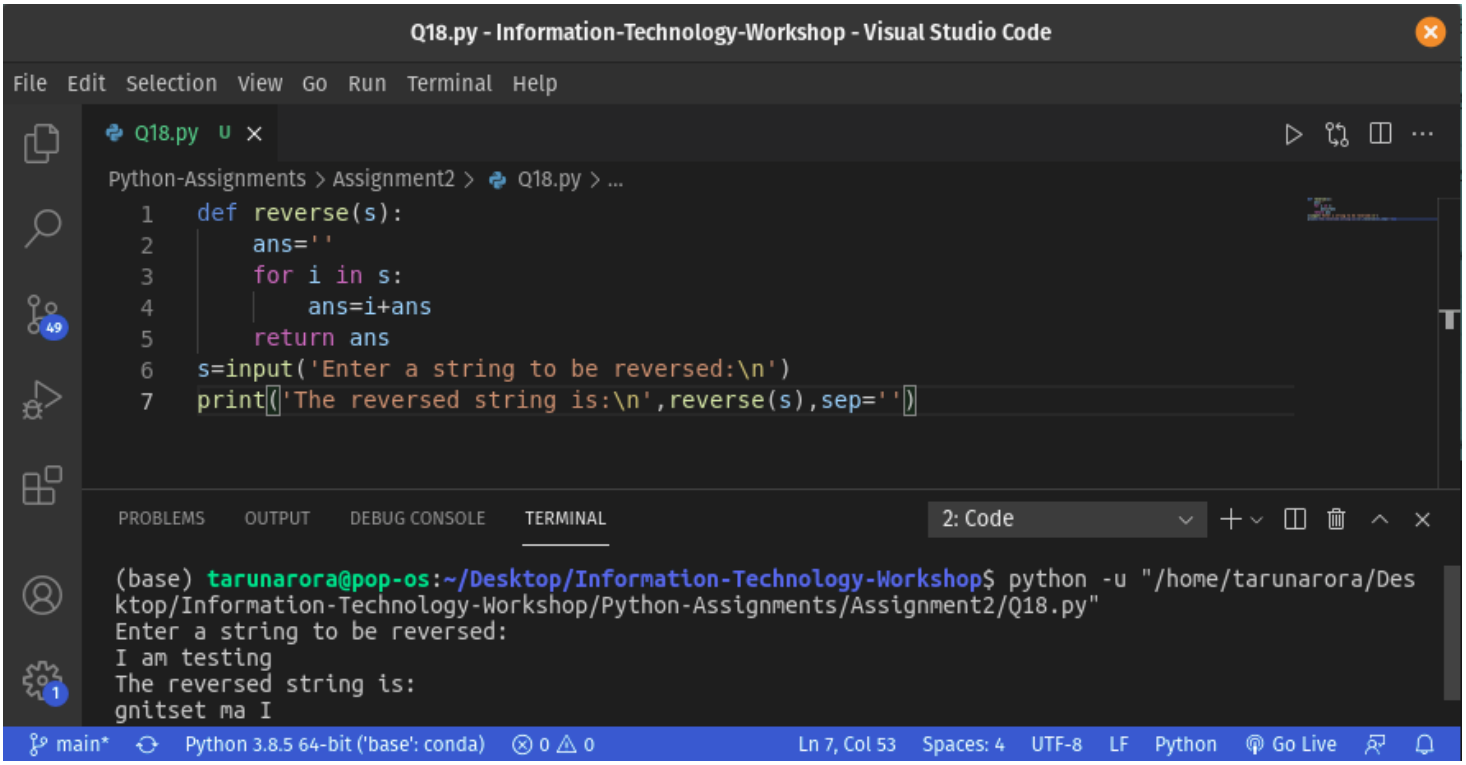
For example, say I type the string: My name is Michele; Then I would see the string: Michele is name My; shown back to me.

Solution: -



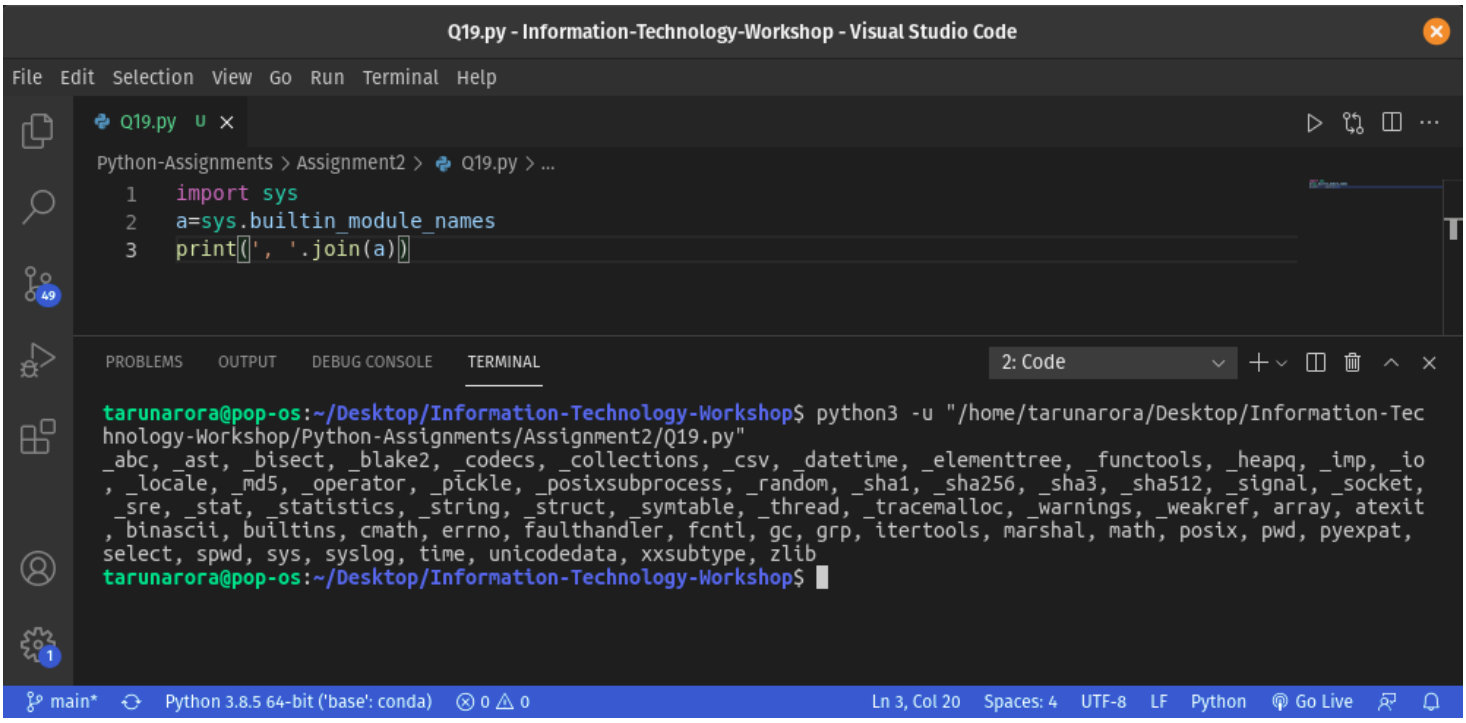
18. Define a function reverse() that computes the reversal of a string. For example, reverse("I am testing") should return the string "gnitset ma I".

Solution: -



19. **Write a Python program to find the available built-in modules. Example: math, random, uuid, sys, syslog etc.**

Solution: -



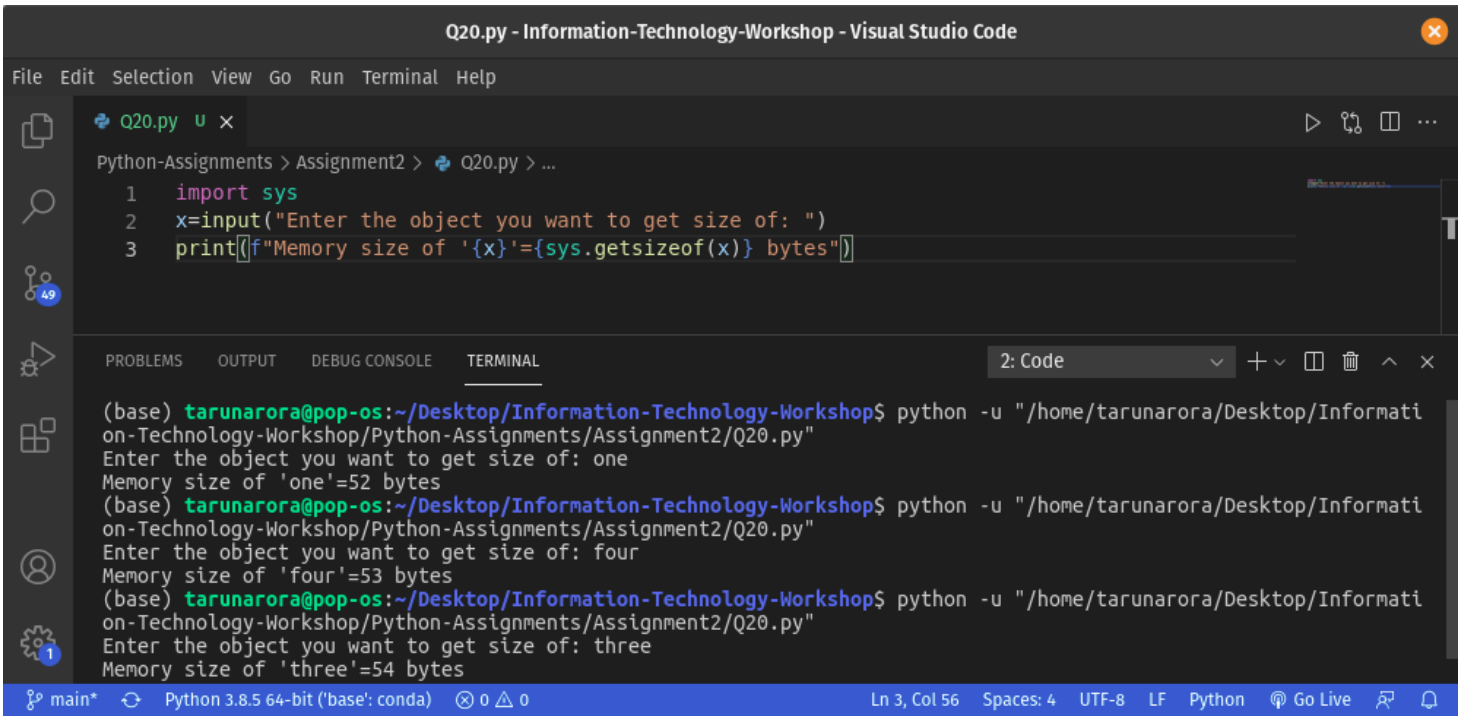
20. Write a Python program to get the size of an object in bytes by using module “sys”.

Example: Memory size of 'one' = 52 bytes

Memory size of 'four' = 53 bytes

Memory size of 'three' = 54 bytes

Solution: -



21. Using the module random and time in python generate a random date between given start and end dates.

Example: Printing random date between 1/1/2016 and 3/23/2018

Random Date = 02/25/2016

Solution: -

Q21.py - Assignment2 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

Q21.py U

Q21.py > ...

```python
import random,time
stDate=input("Enter start date strictly in MM/DD/YYYY format:\n")
try:
    stTime=time.mktime(time.strptime(stDate,'%m/%d/%Y'))
except:
    print('Enter Start Date in correct format\nQuiting Program')
    quit()
enDate=input("Enter end date strictly in MM/DD/YYYY format:\n")
try:
    enTime=time.mktime(time.strptime(enDate,'%m/%d/%Y'))
except:
    print('Enter End Date in correct format\nQuiting Program')
    quit()
pTime=stTime+(random.random())*(enTime-stTime)
if(enTime<stTime):
    print('End Date should be greater than Starting Date')
    quit()
print(f'Printing random date between {stDate} and {enDate}')
print(time.strftime("%m/%d/%Y",time.localtime(pTime)))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Code

```
(base) tarunarora@pop-os:~/Desktop/Information-Technology-Workshop/Python-Assignments/Assignment2$ python -u "/home/tarunarora/Desktop/Information-Technology-Workshop/Python-Assignments/Assignment2/Q21.py"
Enter start date strictly in MM/DD/YYYY format:
3/22/2016
Enter end date strictly in MM/DD/YYYY format:
03/04/2022
Printing random date between 3/22/2016 and 03/04/2022
05/13/2021
(base) tarunarora@pop-os:~/Desktop/Information-Technology-Workshop/Python-Assignments/Assignment2$
```

main* Python 3.8.6 64-bit 0 0 Ln 16, Col 59 Spaces: 4 UTF-8 LF Python

22. Generate three random password string of length 10 with special characters, letters, and digits by using python modules (random and string).

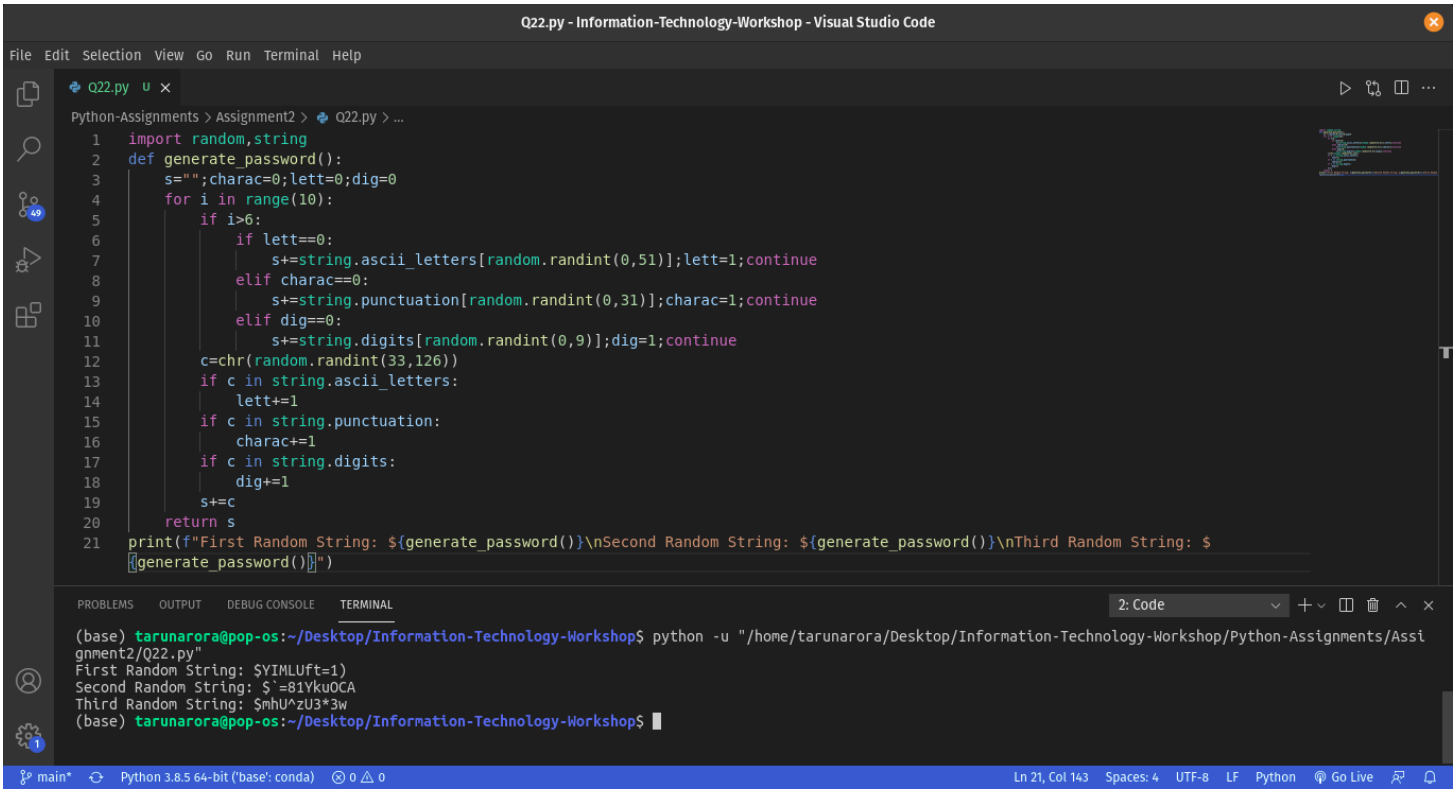
Example:

First Random String: yrjmcyi^VS

Second Random String: |}Hd]!^>~l

Third Random String: 3^a93@x=|Z

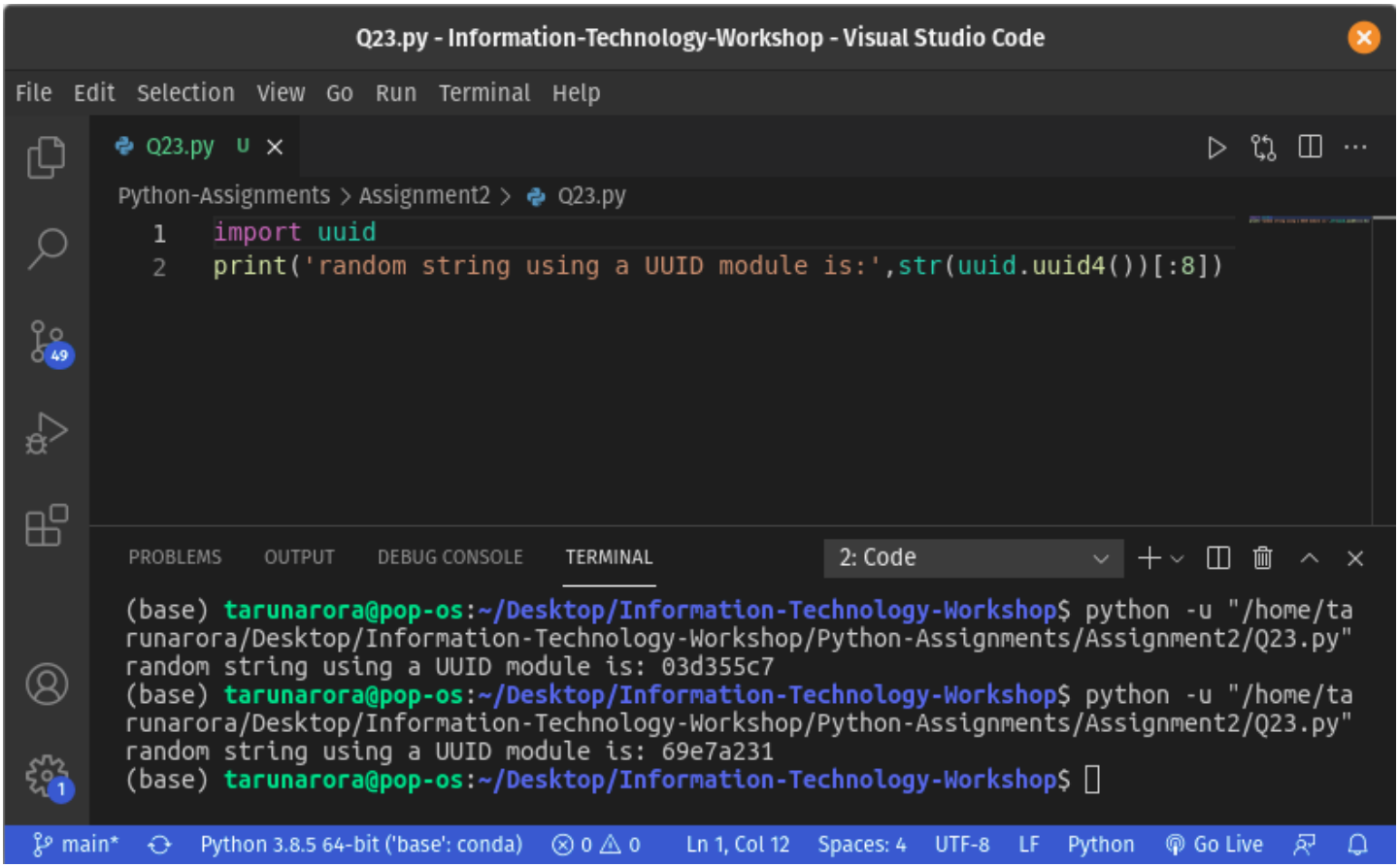
Solution: -



23. Write a python code using module "uuid" to generate universally unique secure randon string id of length 8.

Example: random string using a UUID module is: 9C8E13FF

random string using a UUID module is: 9cb3561d



24. Write a python code using module "random" to generate a 100 Lottery tickets and pick two lucky tickets from it as a winner.

Note: You must adhere to the following conditions:

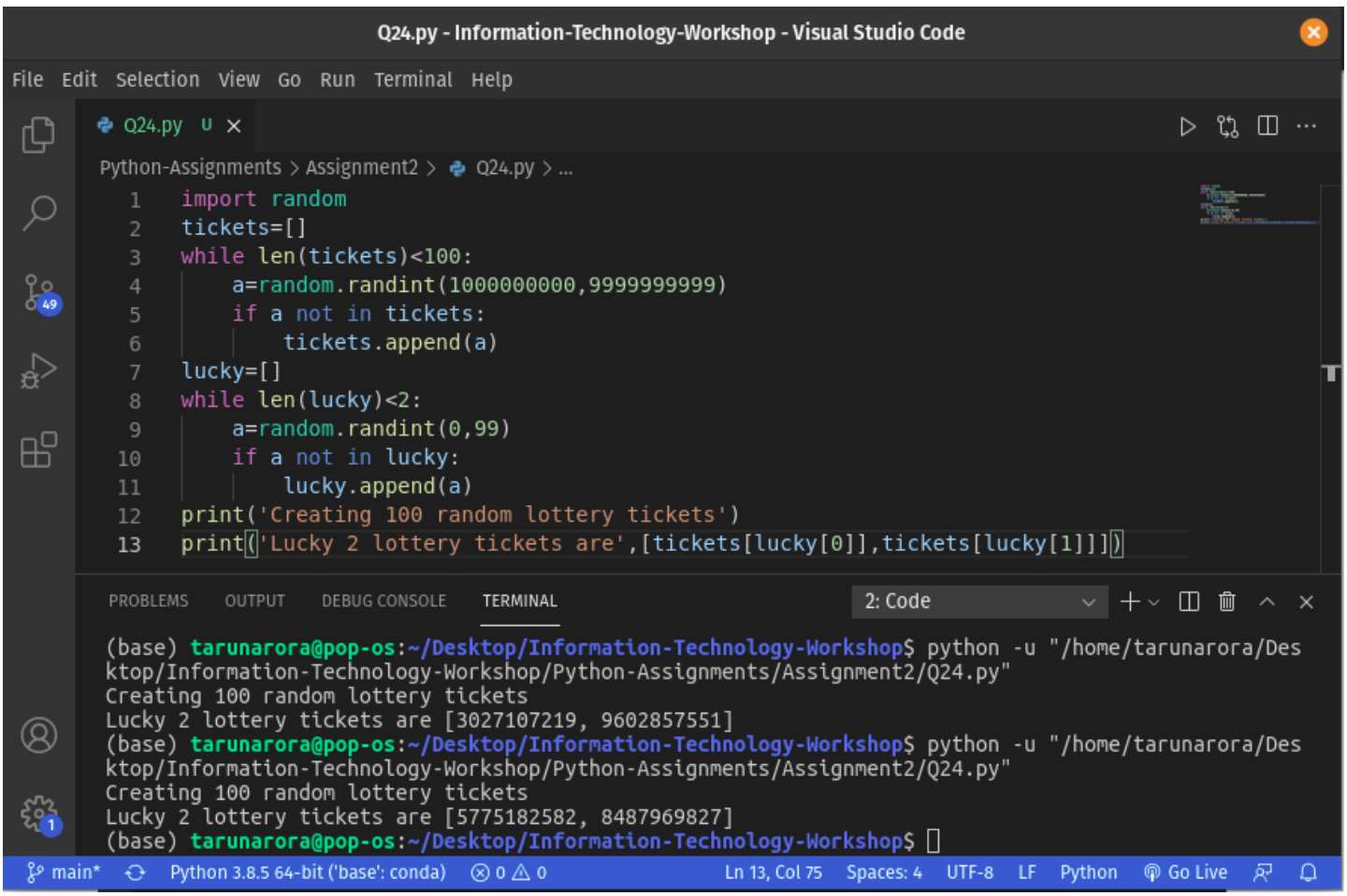
1. Lottery number must be 10 digits long.

2. All 100 ticket number must be unique.

Example: Creating 100 random lottery tickets

Lucky 2 lottery tickets are [7184805696, 7380986204]

Solution: -



********** EOF **********