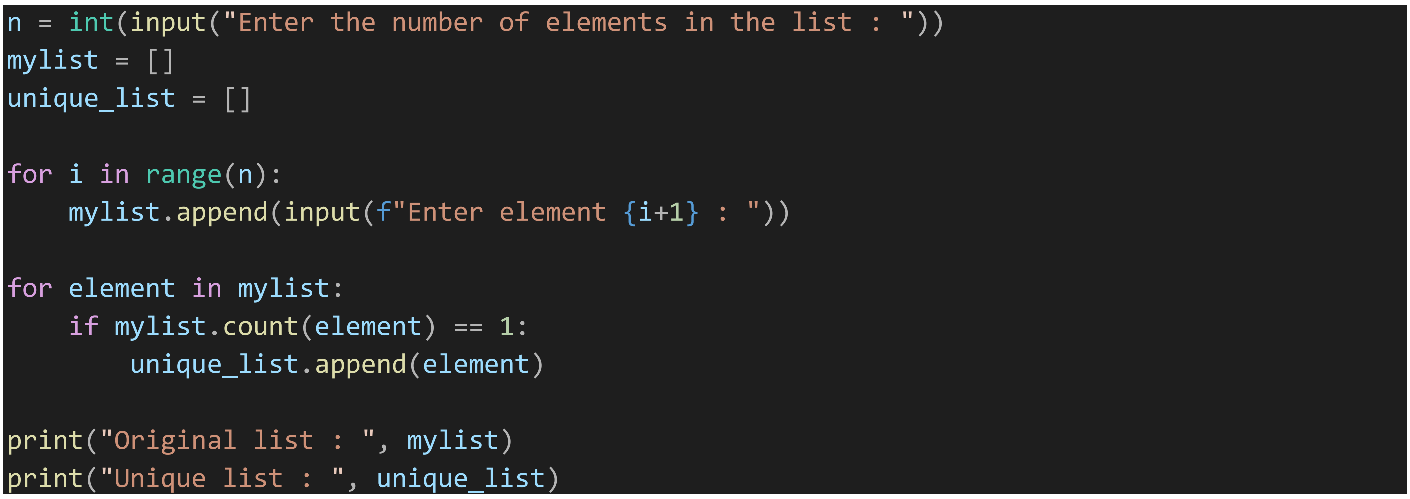
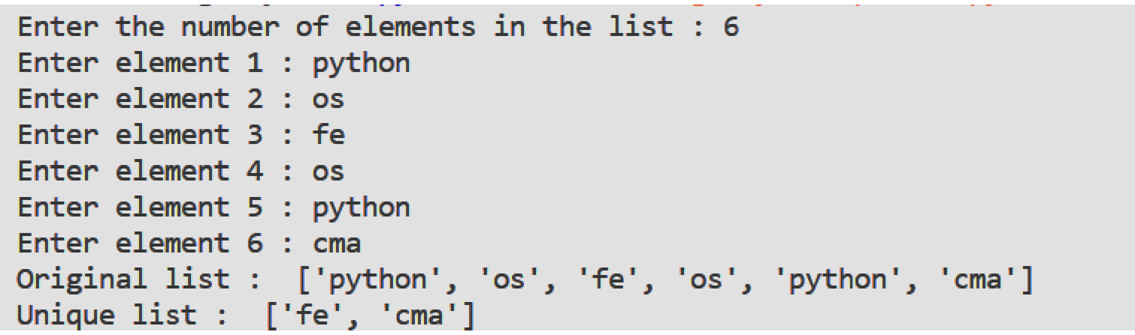
**1.Write a program create list with N elements. find all unique elements in the list. If an element is found only once in the list, then add that element to the unique list.**

## Code :



```
n = int(input("Enter the number of elements in the list : "))
mylist = []
unique_list = []

for i in range(n):
    mylist.append(input(f"Enter element {i+1} : "))

for element in mylist:
    if mylist.count(element) == 1:
        unique_list.append(element)

print("Original list : ", mylist)
print("Unique list : ", unique_list)
```

## Output :

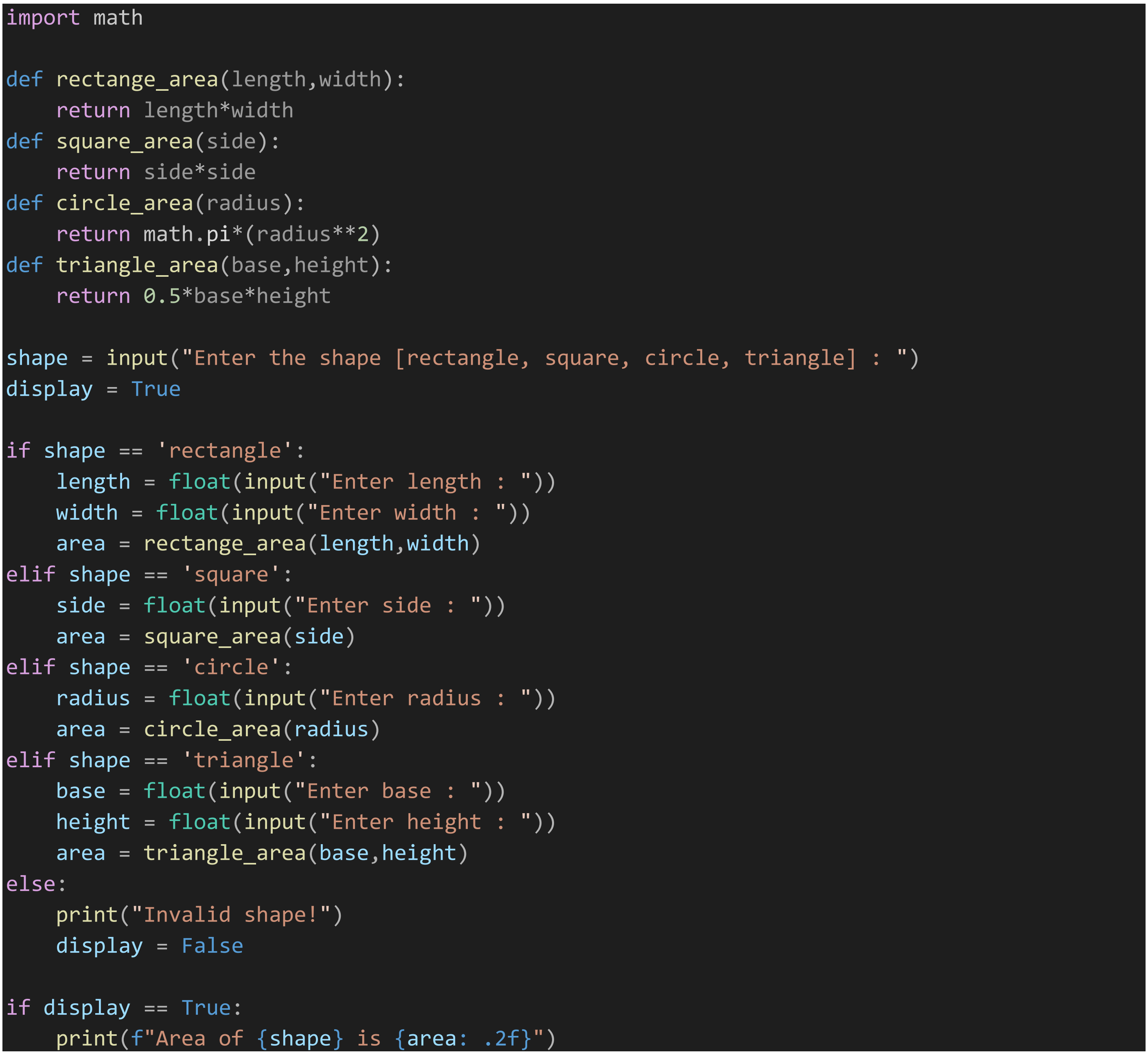


```
Enter the number of elements in the list : 6
Enter element 1 : python
Enter element 2 : os
Enter element 3 : fe
Enter element 4 : os
Enter element 5 : python
Enter element 6 : cma
Original list :  ['python', 'os', 'fe', 'os', 'python', 'cma']
Unique list :  ['fe', 'cma']
```

-ABR

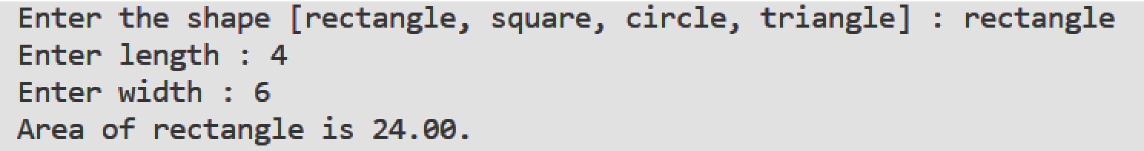
## 2.Program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.

### Code :

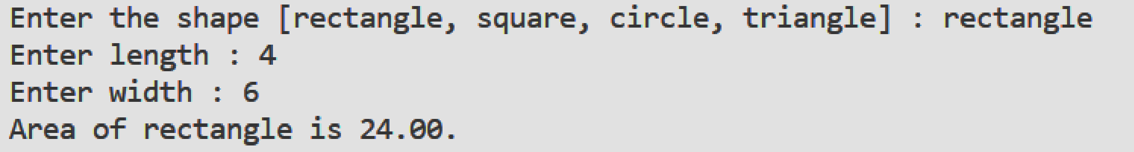


```python
import math

def rectange_area(length,width):
    return length*width
def square_area(side):
    return side*side
def circle_area(radius):
    return math.pi*(radius**2)
def triangle_area(base,height):
    return 0.5*base*height

shape = input("Enter the shape [rectangle, square, circle, triangle] : ")
display = True

if shape == 'rectangle':
    length = float(input("Enter length : "))
    width = float(input("Enter width : "))
    area = rectange_area(length,width)
elif shape == 'square':
    side = float(input("Enter side : "))
    area = square_area(side)
elif shape == 'circle':
    radius = float(input("Enter radius : "))
    area = circle_area(radius)
elif shape == 'triangle':
    base = float(input("Enter base : "))
    height = float(input("Enter height : "))
    area = triangle_area(base,height)
else:
    print("Invalid shape!")
    display = False

if display == True:
    print(f"Area of {shape} is {area: .2f}")
```

### Output :

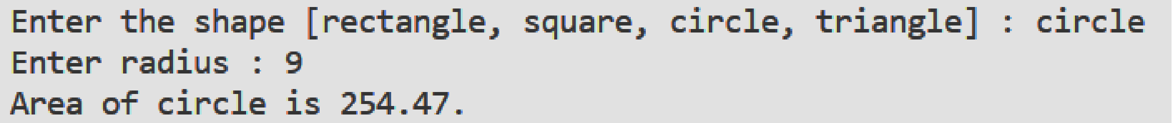


```
Enter the shape [rectangle, square, circle, triangle] : rectangle
Enter length : 4
Enter width : 6
Area of rectangle is 24.00.
```

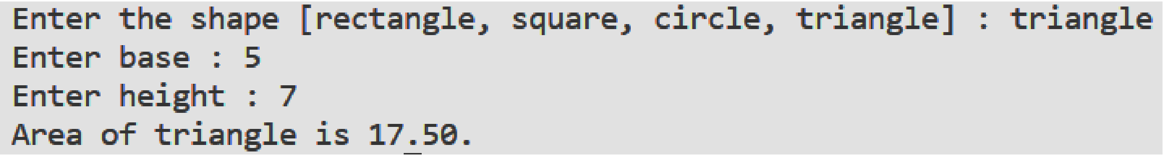


```
Enter the shape [rectangle, square, circle, triangle] : rectangle
Enter length : 4
Enter width : 6
Area of rectangle is 24.00.
```

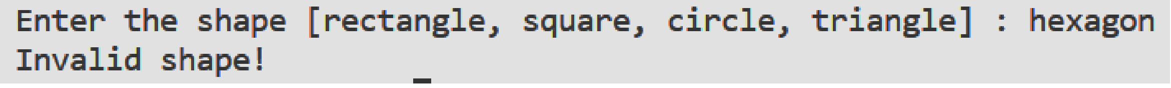
-ABR



```
Enter the shape [rectangle, square, circle, triangle] : circle
Enter radius : 9
Area of circle is 254.47.
```



```
Enter the shape [rectangle, square, circle, triangle] : triangle
Enter base : 5
Enter height : 7
Area of triangle is 17.50.
```

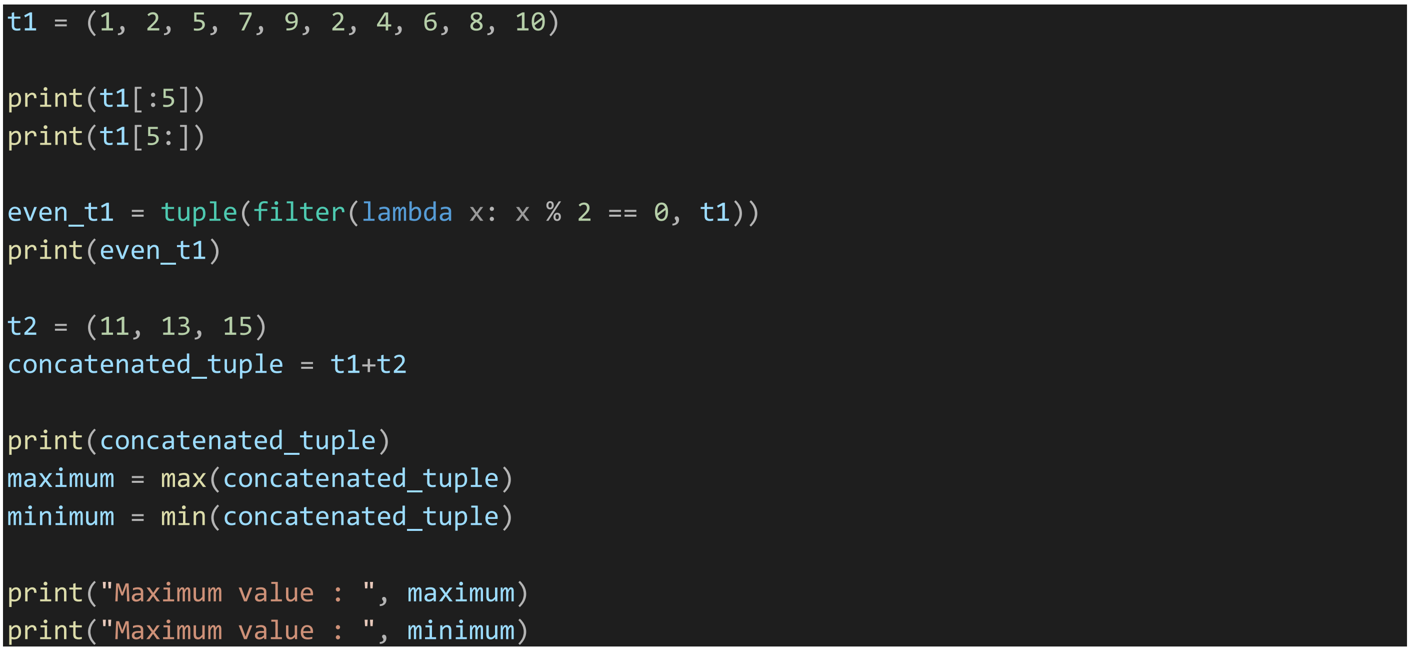


```
Enter the shape [rectangle, square, circle, triangle] : hexagon
Invalid shape!
```

3. Consider a tuple t1= (1,2,5,7,9,2,4,6,8,10). Write a program to perform following operations:

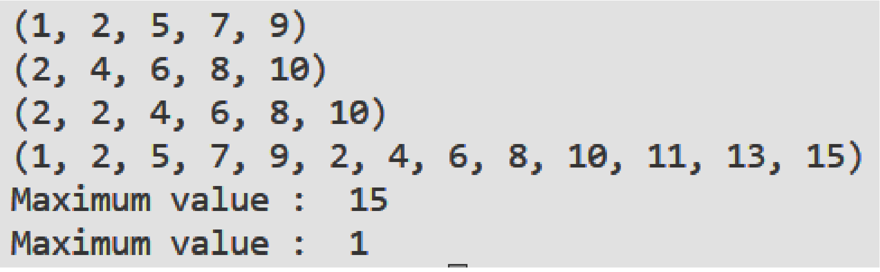
   a. Print half the values of tuple in one line and the other half in the next line.
   b. Print another tuple whose values are even numbers in the given tuple.
   c. Concatenate a tuple t2= (11,13,15) with t1.
   d. Return maximum and minimum value from this tuple.

**Code:**



```
t1 = (1, 2, 5, 7, 9, 2, 4, 6, 8, 10)

print(t1[:5])
print(t1[5:])

even_t1 = tuple(filter(lambda x: x % 2 == 0, t1))
print(even_t1)

t2 = (11, 13, 15)
concatenated_tuple = t1+t2

print(concatenated_tuple)
maximum = max(concatenated_tuple)
minimum = min(concatenated_tuple)

print("Maximum value : ", maximum)
print("Maximum value : ", minimum)
```

**Output :**

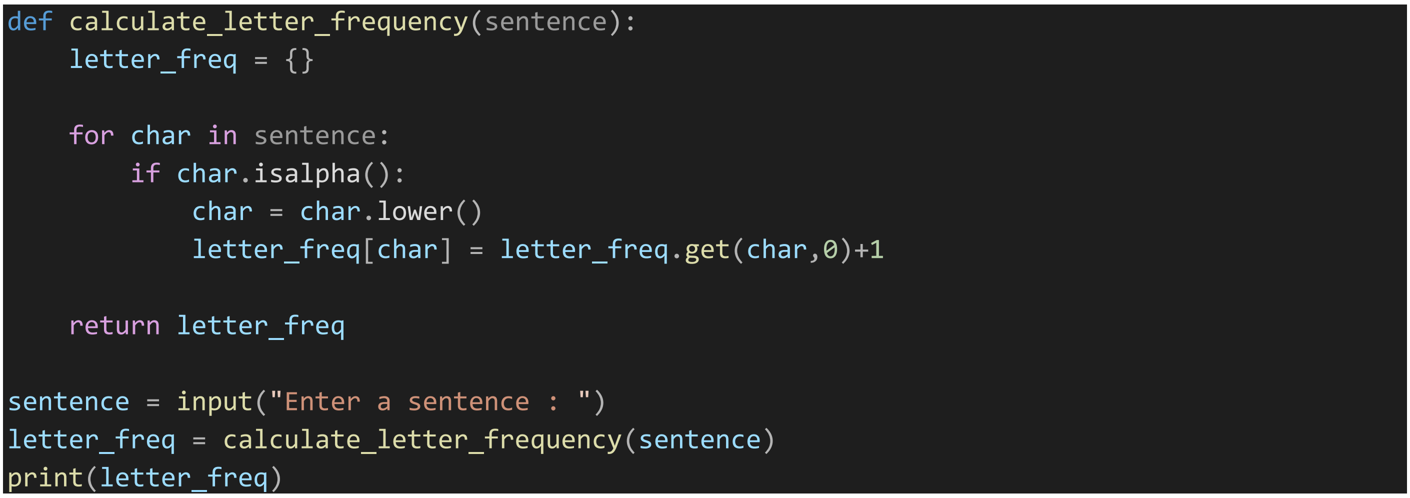


```
(1, 2, 5, 7, 9)
(2, 4, 6, 8, 10)
(2, 2, 4, 6, 8, 10)
(1, 2, 5, 7, 9, 2, 4, 6, 8, 10, 11, 13, 15)
Maximum value :  15
Maximum value :  1
```

-ABR

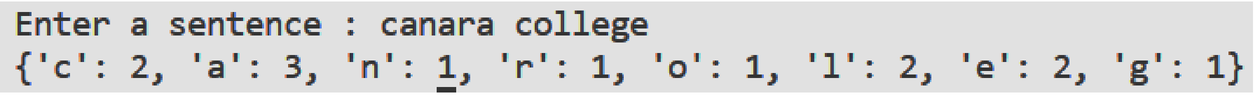
**4.Write a function that takes a sentence as input from the user and calculates the frequency of each letter. Use a variable of dictionary type to maintain the count.**

## Code :



```
def calculate_letter_frequency(sentence):
    letter_freq = {}

    for char in sentence:
        if char.isalpha():
            char = char.lower()
            letter_freq[char] = letter_freq.get(char,0)+1

    return letter_freq

sentence = input("Enter a sentence : ")
letter_freq = calculate_letter_frequency(sentence)
print(letter_freq)
```

**Output :**

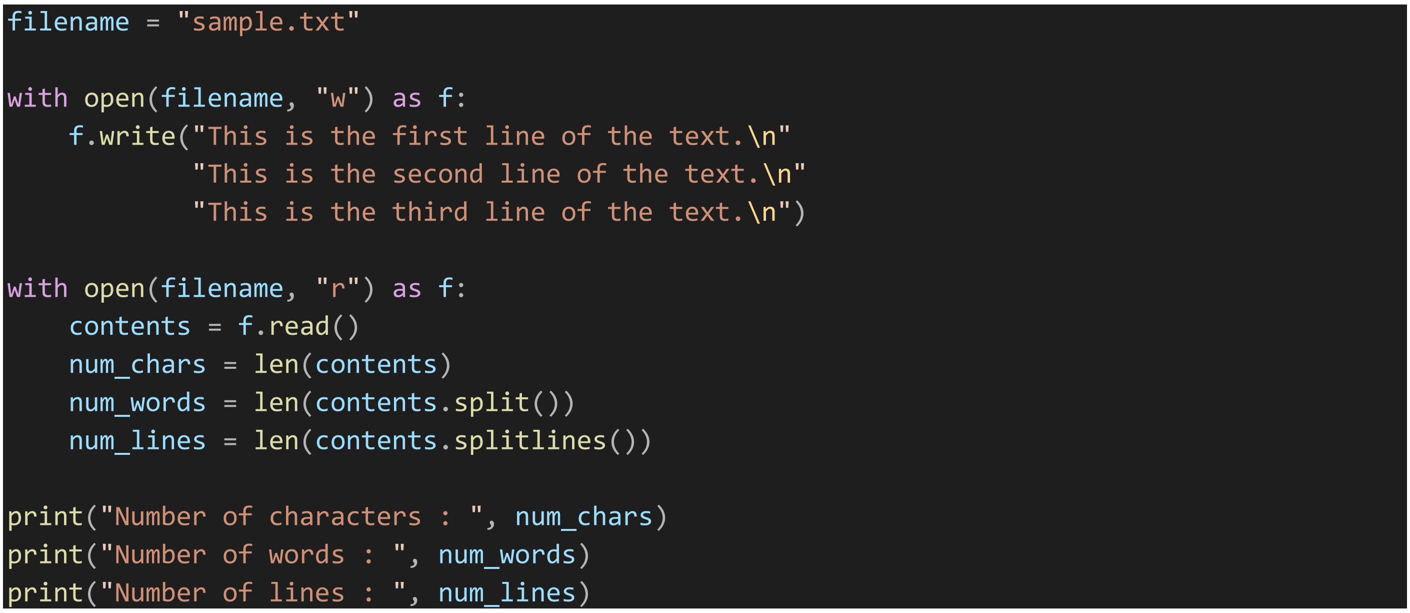


```
Enter a sentence : canara college
{'c': 2, 'a': 3, 'n': 1, 'r': 1, 'o': 1, 'l': 2, 'e': 2, 'g': 1}
```

-ABR

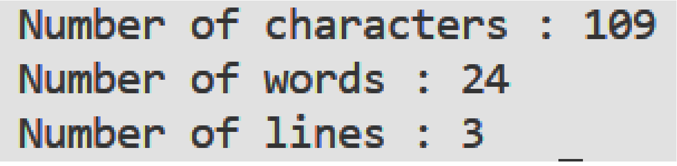
## 6.Write a program to create a text file and compute the number of characters, words and lines in a file.

### Code :



```
filename = "sample.txt"

with open(filename, "w") as f:
    f.write("This is the first line of the text.\n"
            "This is the second line of the text.\n"
            "This is the third line of the text.\n")

with open(filename, "r") as f:
    contents = f.read()
    num_chars = len(contents)
    num_words = len(contents.split())
    num_lines = len(contents.splitlines())

print("Number of characters : ", num_chars)
print("Number of words : ", num_words)
print("Number of lines : ", num_lines)
```

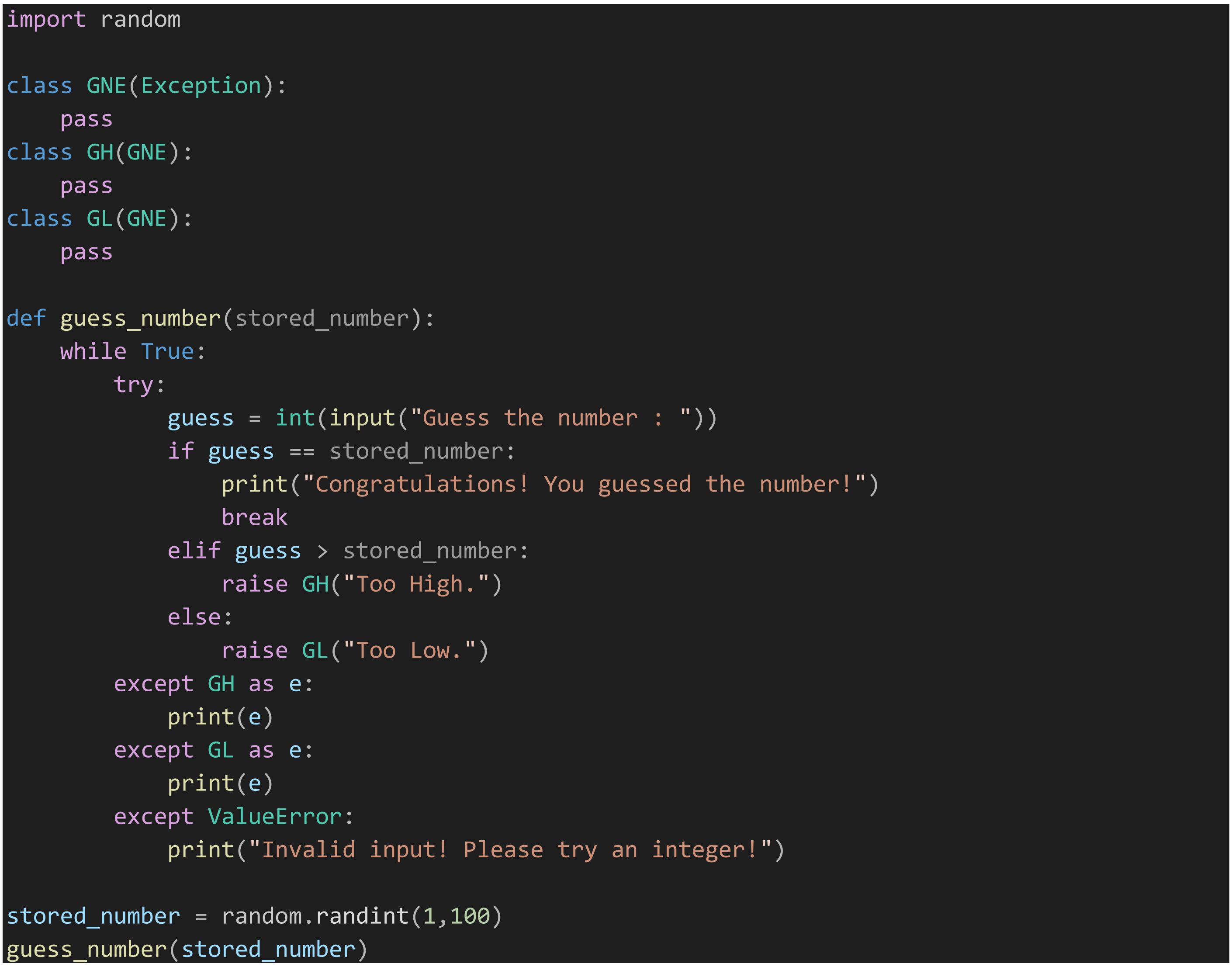
### Output:



```
Number of characters : 109
Number of words : 24
Number of lines : 3
```

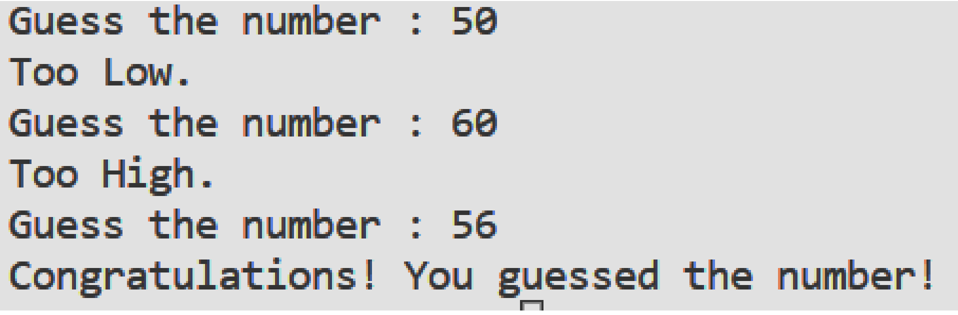
7. Program using user defined exception class that will ask the user to enter a number until he guesses a stored number correctly. To help them figure it out, a hint is provided whether their guess is greater than or less than the stored number using user defined exceptions.

## Code :



```python
import random

class GNE(Exception):
    pass
class GH(GNE):
    pass
class GL(GNE):
    pass

def guess_number(stored_number):
    while True:
        try:
            guess = int(input("Guess the number : "))
            if guess == stored_number:
                print("Congratulations! You guessed the number!")
                break
            elif guess > stored_number:
                raise GH("Too High.")
            else:
                raise GL("Too Low.")
        except GH as e:
            print(e)
        except GL as e:
            print(e)
        except ValueError:
            print("Invalid input! Please try an integer!")

stored_number = random.randint(1,100)
guess_number(stored_number)
```

## Output :



```
Guess the number : 50
Too Low.
Guess the number : 60
Too High.
Guess the number : 56
Congratulations! You guessed the number!
```

-ABR

## Source Code :

https://drive.google.com/drive/folders/1wgbz-b0JK_8LitS4M5KzIZ7swAMi2_t4?usp=sharing

-ABR