

University of Engineering and Management (UEM),Kolkata

Department of Computer Applications

Stream: MCA

Session: 2024-2026

PPT Assignment-3

Subject Name: Pre Placement Training

Name:-Atreyee Chakraborty

Enrollment number:-12024006015068

Roll:-12 ,Sec:-B

Q-1. Write a program to input a 4 digit number and print the even and odd digits and total even and odd digits

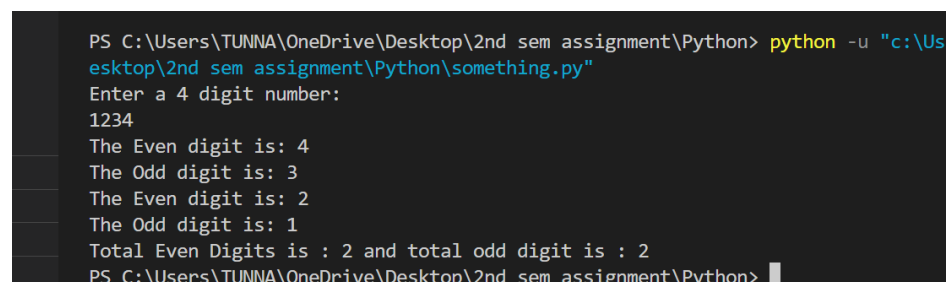
```
def even_odd(n):  
    even=0  
    odd=0  
    while n>0:  
        digit=n%10  
        if digit%2!=0:  
            print("The Odd digit is: {digit}")  
            odd+=1  
        else:  
            print("The Even digit is: {digit}")
```

```

        even+=1
    n=n//10
    print("Total Even Digits is : {even} and total odd digit is : {odd}")
Number=int(input("Enter a 4 digit number:\n"))
even_odd(Number)

```

OUTPUT:-



```

PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
Enter a 4 digit number:
1234
The Even digit is: 4
The Odd digit is: 3
The Even digit is: 2
The Odd digit is: 1
Total Even Digits is : 2 and total odd digit is : 2
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>

```

Q-2. Write a program to input a 10 digit number and find the digit with maximum value. +

```

def Enter_ten_Digits(n):
    seq=[]
    while n>0:
        digit=n%10
        seq.append(digit)
        n=n//10
    seq.sort()
    print("The maximum Digit is {seq[-1]}")
Number=int(input("Enter a 10 digit number:\n"))
Enter_ten_Digits(Number)

```

OUTPUT:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE Code + - [ ] [ ] ... ^ X

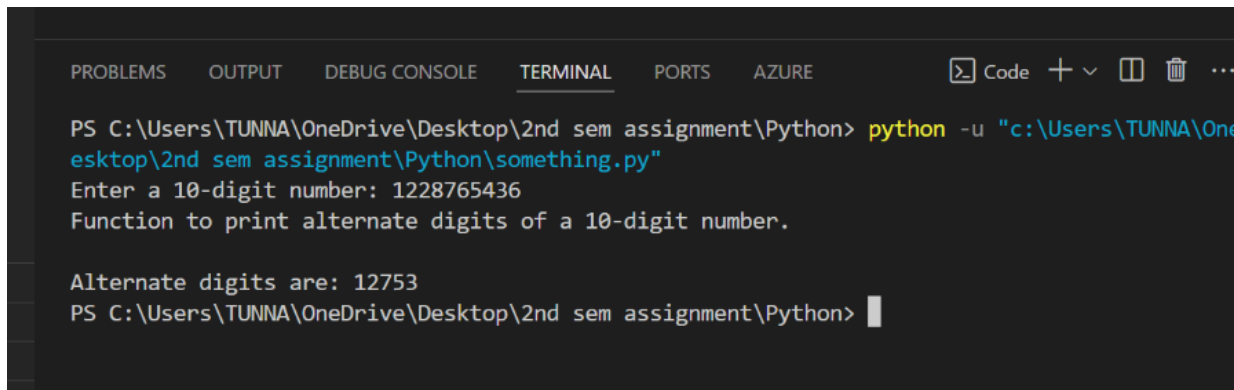
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
Enter a 10 digit number:
1234567891
The maximum Digit is 9
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> |
```

3. Write a program to input a 10 digit number and print alternate digits.

```
def print_alternate_digits(number):
    """Function to print alternate digits of a 10-digit number.
    """
    alternate_digits = number[::2]
    print("Alternate digits are:", alternate_digits)

def main():
    number = input("Enter a 10-digit number: ")
    if len(number) == 10 and number.isdigit():
        print(print_alternate_digits.__doc__)
        print_alternate_digits(number)
    else:
        print("Invalid input! Please enter exactly a 10-digit number.")
main()
```

OUTPUT:-

A screenshot of a Windows terminal window. The title bar shows tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (active), PORTS, and AZURE. The terminal text shows a command prompt at PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> running python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py". The program prompts "Enter a 10-digit number: 1228765436" and then prints "Function to print alternate digits of a 10-digit number." followed by "Alternate digits are: 12753". The prompt returns to PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>.

```
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
Enter a 10-digit number: 1228765436
Function to print alternate digits of a 10-digit number.

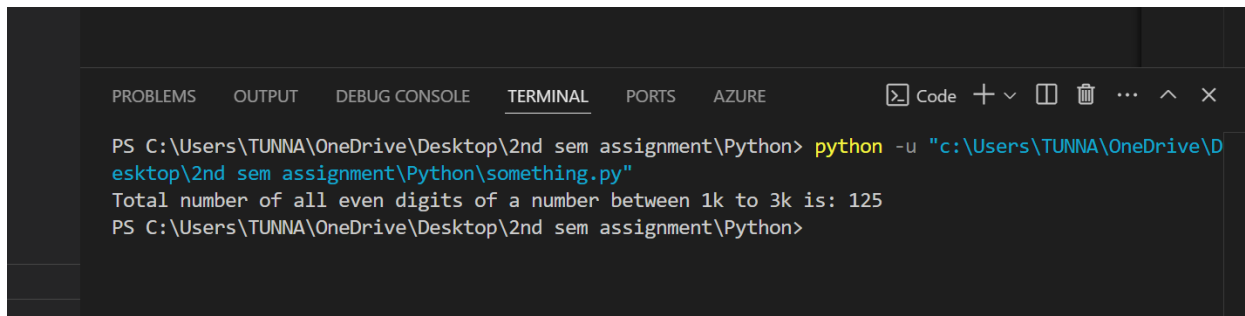
Alternate digits are: 12753
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>
```

4. Write a program, which will find all such numbers between 1000 and 3000 such that each digit of the number is an even number.

```
def isEvedigits(n):
    for i in str(n):
        if int(i)%2!=0:
            return False
    return True

even=[ ]
for i in range(1001,3000):
    if isEvedigits(i):
        even.append(i)
        #print("The number {i} is even.")
print("Total number of all even digits of a number between 1k to 3k is:
{len(even)}")
```

OUTPUT:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
Total number of all even digits of a number between 1k to 3k is: 125
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>
```

5. Write a program that prints a list where the values are square of numbers between 5000 and 7000 (both included).

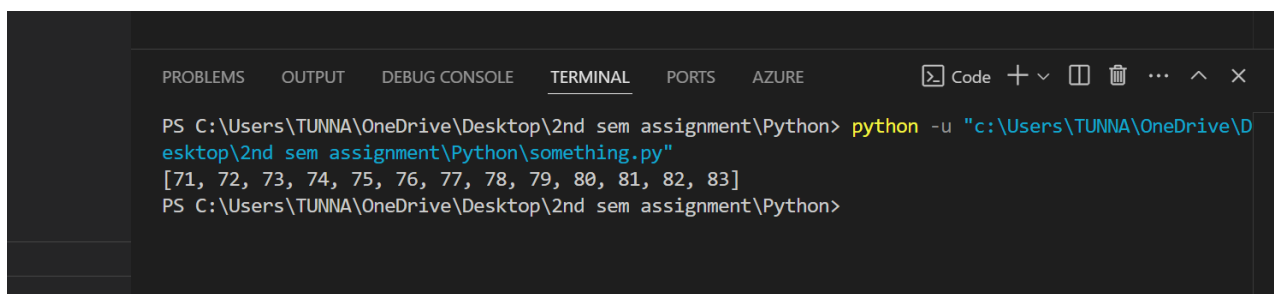
```
import math

square_values=[]

for i in range(5000,7001):
    sqr_num=math.isqrt(i)
    if sqr_num**2==i:
        square_values.append(sqr_num)
    else:
        continue

print(square_values)
```

OUTPUT:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
[71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83]
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>
```

6. Given a sorted array of positive integers arr, and an integer n which represents the length of arr, the task is to

rearrange the array elements alternatively i.e first element should be max value, second should be min value, third should be second max, fourth should be second min and so on.

```
def rearrange_Array(arr,n):  
    result=[]  
    min=0  
    max=n-1  
    for inx in range(1,n+1):  
        if inx%2==0:  
            result.append(arr[min])  
            min+=1  
        else:  
            result.append(arr[max])  
            max-=1  
    print("Rearranged Array:", result)  
arr = [1, 2, 3, 4, 5, 6,7]  
n = len(arr)  
rearrange_Array(arr, n)
```

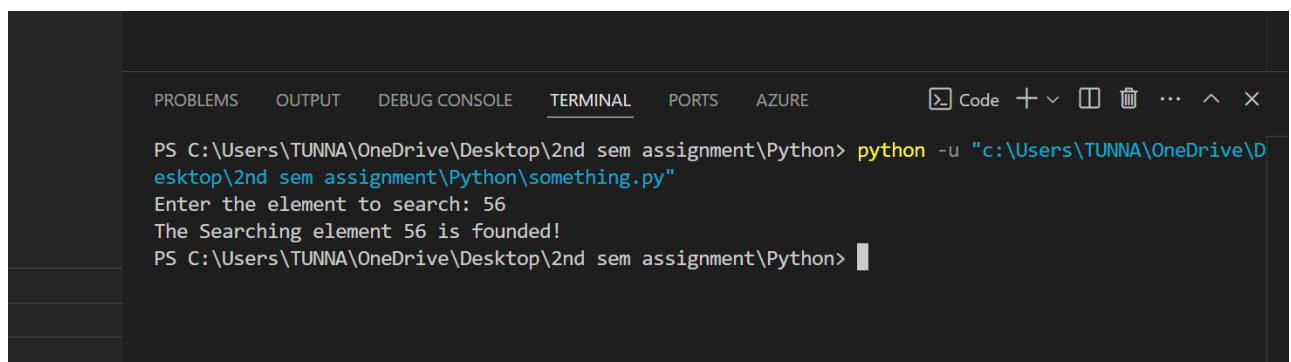
OUTPUT:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  AZURE  Code + - [] [X] ... ^ X  
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"  
Rearranged Array: [7, 1, 6, 2, 5, 3, 4]  
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>
```

7. Write a program to input a number and search in a list using linear search.

```
def isPresent(arr,n):  
    m=len(arr)  
    flag=0  
    for i in range(m):  
        if arr[i]==n:  
            flag=1  
        else:  
            continue  
    if flag==1:  
        print("The Searching element {n} is founded!")  
    else:  
        print("The Searching element is not founded.")  
arr = [34,45,56,67,78,89,90]  
search=int(input("Enter the element to search: "))  
isPresent(arr, search)
```

OUTPUT:-



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), 'PORTS', and 'AZURE'. Below the tabs, the command prompt shows the execution of a Python script: `PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"`. The program then prompts for input: `Enter the element to search: 56`. The output of the program is: `The Searching element 56 is founded!`. The prompt returns to the command line: `PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>`.

8. Write a program to input a number and search in a list using binary search.

```

def binary_Search(arr,target):
    beg=0
    end=len(arr)-1
    while beg<=end:
        mid=(beg+end)//2
        if arr[mid]==target:
            return 1
        elif arr[mid]<target:
            beg=mid+1
        else:
            end=mid-1
    return -1

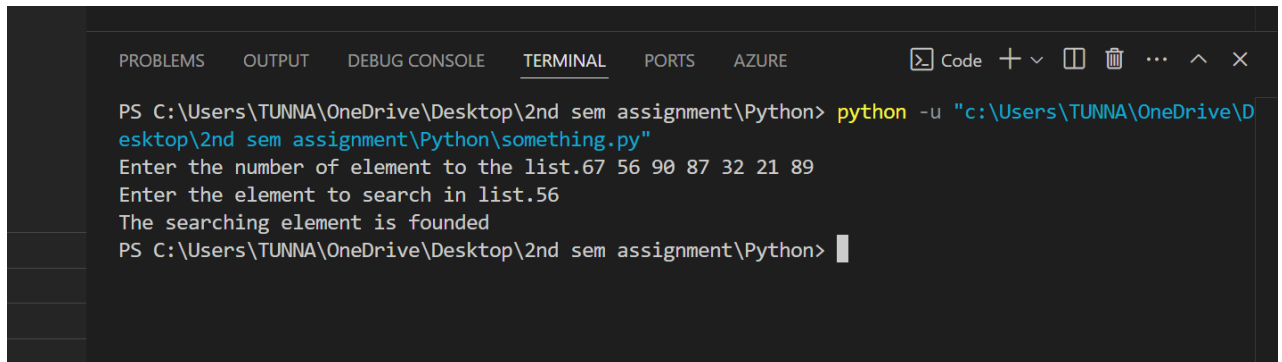
def main():
    try:
        user_list=list(map(int,input("Enter the number of element to the list.").split()))
        user_list.sort()
        Search=int(input("Enter the element to search in list. "))
        result=binary_Search(user_list,Search)

        if result==1:
            print("The searching element is founded")
        else:
            print("The searching element is not founded.")
    except ValueError:
        print("Invalid input! Please enter a valid integer.")

```


main()

OUTPUT:-



```
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
Enter the number of element to the list.67 56 90 87 32 21 89
Enter the element to search in list.56
The searching element is founded
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> 
```

9. Write a program to input a list of numbers and sort using bubble sort.

```
def bubble_Sorting(arr):
    n=len(arr)
    flag=False
    for i in range(n-1):
        for j in range(n-i-1):
            if arr[j]>arr[j+1]:
                arr[j],arr[j+1]=arr[j+1],arr[j]
                flag=True
        if not flag:
            break
def main():
    try:
        array=list(map(int,input("Enter the elements of the list.").split()))
        print("Before the bubble sort the list is {array}")
```

```

bubble_Sorting(array)

print("After the bubble sort the list is {array}")

except ValueError:

    print("invalid input! Please enter a valid ineger.")

main()

```

OUTPUT:-

```

PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
Enter the elements of the list.33 90 12 78 34 67 25 89 43 76
Before the bubble sort the list is [33, 90, 12, 78, 34, 67, 25, 89, 43, 76]
After the bubble sort the list is [12, 25, 33, 34, 43, 67, 76, 78, 89, 90]
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>

```

10. Write a program to input a list of numbers and sort using insertion sort.

```

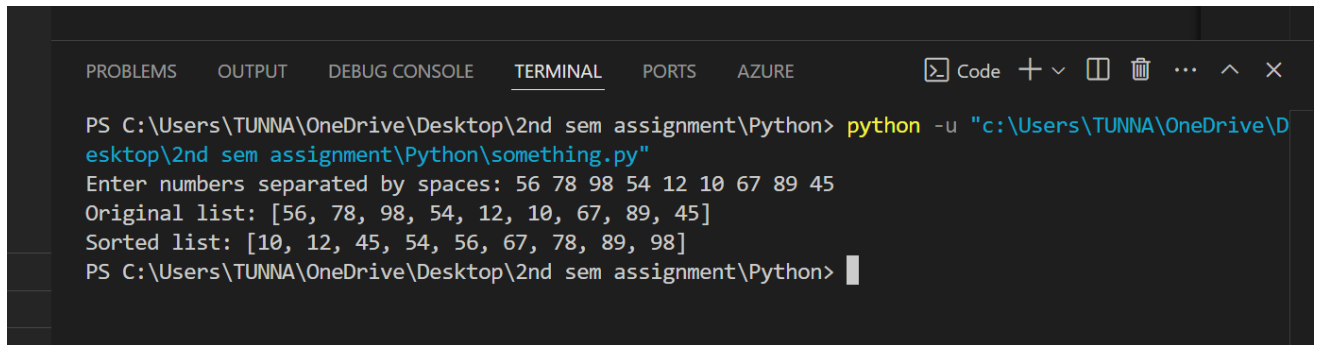
def insertion_sort(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key

numbers = list(map(int, input("Enter numbers separated by spaces: ").split()))

```

```
print("Original list:", numbers)
insertion_sort(numbers)
print("Sorted list:", numbers)
```

OUTPUT:-



```
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
Enter numbers separated by spaces: 56 78 98 54 12 10 67 89 45
Original list: [56, 78, 98, 54, 12, 10, 67, 89, 45]
Sorted list: [10, 12, 45, 54, 56, 67, 78, 89, 98]
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> 
```

11. Write a program to input a list of numbers and sort using selection sort.

```
def selection_Sorting(arr):
    n=len(arr)
    for i in range(n):
        min_idx=i
        for j in range(i+1,n):
            if arr[j]<arr[min_idx]:
                min_idx=j
        arr[i],arr[min_idx]=arr[min_idx],arr[i]

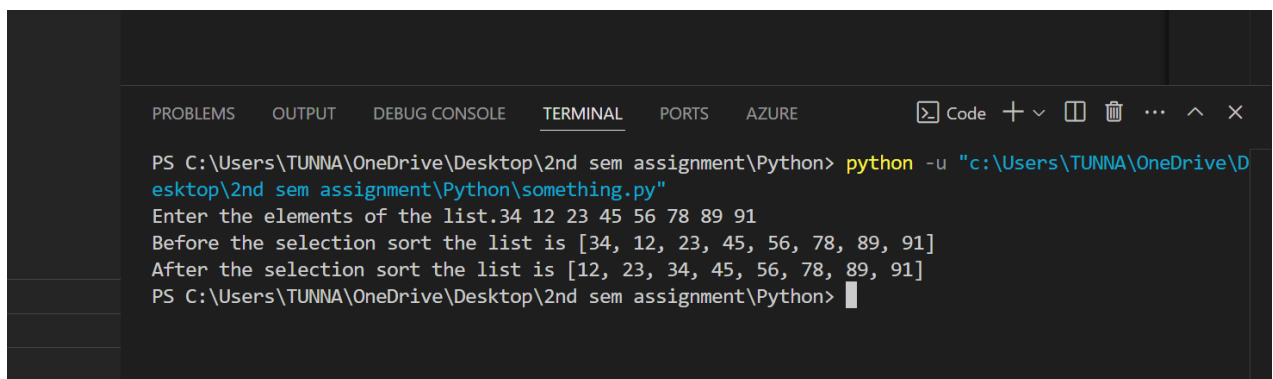
def main():
    try:
        array=list(map(int,input("Enter the elements of the list.").split()))
        print("Before the selection sort the list is {array}")
```

```

        selection_Sorting(array)
    print("After the selection sort the list is {array}")
except ValueError:
    print("invalid input! Please enter a valid ineger.")
main()

```

OUTPUT:-



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
Enter the elements of the list.34 12 23 45 56 78 89 91
Before the selection sort the list is [34, 12, 23, 45, 56, 78, 89, 91]
After the selection sort the list is [12, 23, 34, 45, 56, 78, 89, 91]
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>

```

12. Write a program to find the frequency of characters in a given string.

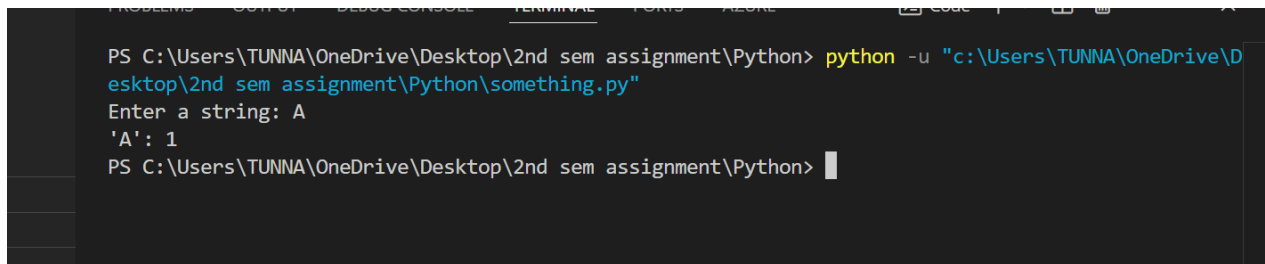
```

def frequency_characters(input_string):
    freq_dict={}
    for i in input_string:
        if i in freq_dict:
            freq_dict[i]+=1
        else:
            freq_dict[i]=1
    return freq_dict
input_string=input("Enter a string: ")

```

```
frequencies=frequency_characters(input_string)
for char, value in frequencies.items():
    print("{char}': {value}")
```

OUTPUT:-



```
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
Enter a string: A
'A': 1
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>
```

13. Write a program to input a string and a number to encode the string by adding the number to every character in the string.

```
def encode(string,num):
    seq=""
    for i in string:
        new_char=chr(ord(i)+num)
        seq+=new_char
    print(seq)
string_input=input("Enter a String: ")
number=int(input("Enter a integer no. to add. "))
encode(string_input,number)
```

OUTPUT:-

```
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
Enter a String: Atreyee
Enter a integer no. to add. 19
Txxx
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> |
```

**# 14.Create an acronym or an abbreviation for the name
'Python For Everyone'.**

```
def create_acronym(phrase):
```

```
    words = phrase.split()
```

```
    acronym = ''.join([word[0].upper() for word in words])
```

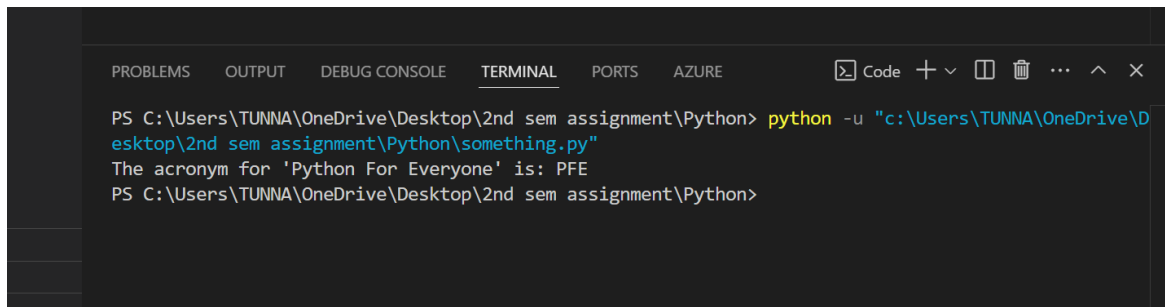
```
    return acronym
```

```
phrase = "Python For Everyone"
```

```
acronym = create_acronym(phrase)
```

```
print("The acronym for '{phrase}' is: {acronym}")
```

OUTPUT:-

A screenshot of a terminal window within a code editor. The terminal shows a command prompt where a Python script is executed. The script prints the acronym for 'Python For Everyone' as 'PFE'.

```
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
The acronym for 'Python For Everyone' is: PFE
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>
```

15. Find the position of the first occurrence of the word 'because' in the following sentence: 'You cannot end a sentence with because because because is a conjunction';

```
def find_first_occurrence(sentence, word):
```

```
    words = sentence.split()
```

```
    for index, current_word in enumerate(words):
```

```
        if current_word.lower() == word.lower():
```

```
            return index + 1
```

```
    return -1
```

```
sentence = "'You cannot end a sentence with because because  
because is a conjunction';"
```

```
word = "because"
```

```
position = find_first_occurrence(sentence, word)
```

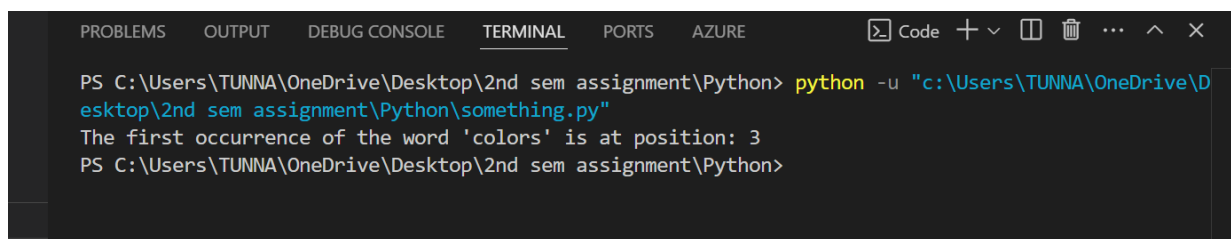
```
if position != -1:
```

```
print("The first occurrence of the word '{word}' is at position:  
{position}")
```

else:

```
print("The word '{word}' was not found in the sentence.")
```

OUTPUT:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE Code + - [ ] [ ] ... ^ X  
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\De  
esktop\2nd sem assignment\Python\something.py"  
The first occurrence of the word 'colors' is at position: 3  
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>
```

16.Slice out the phrase 'because because because' in the following sentence: 'You cannot end a sentence with because because because is a conjunction';

```
def remove_phrase(sentence, phrase):  
    return sentence.replace(phrase, "")
```

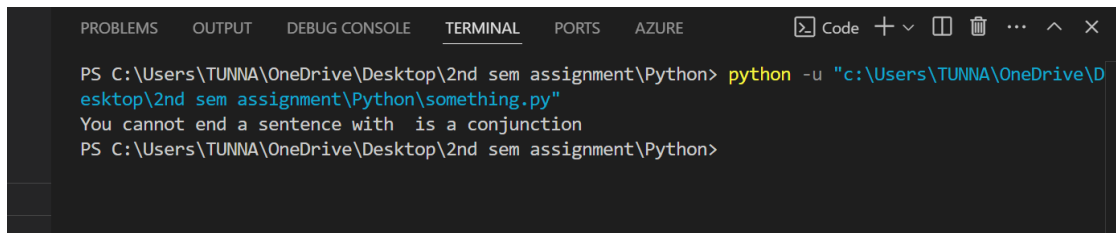
```
sentence = "You cannot end a sentence with because because because is a  
conjunction"
```

```
phrase = "because because because"
```

```
new_sentence = remove_phrase(sentence, phrase)
```

```
print(new_sentence)
```


OUTPUT:-

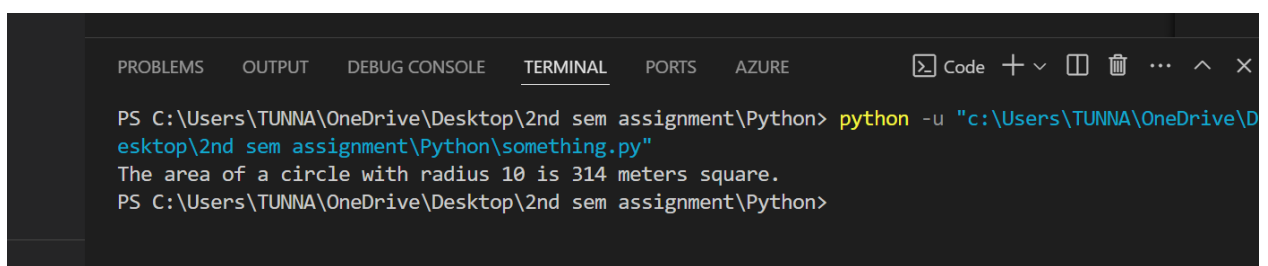


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
You cannot end a sentence with is a conjunction
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>
```

**# 17. Use the string formatting method to display the following:
radius = 10 area = 3.14 * radius ** 2 The area of a circle with
radius 10 is 314 meters square.**

```
def string_formatting(radius,area):
    result="The area of a circle with radius {} is {:.0f} meters
square.".format(radius, area)
    print(result)
def main():
    radius = 10
    area=3.14 * radius **2
    string_formatting(radius,area)
main()
```

OUTPUT:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python> python -u "c:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python\something.py"
The area of a circle with radius 10 is 314 meters square.
PS C:\Users\TUNNA\OneDrive\Desktop\2nd sem assignment\Python>
```

