EXTRA TASK

DATA STRUCTURES

1. Create a list of given structure and get the Access list as provided below: x=[100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]

a)Access list: [1, 2, 3, 4]

# Access by given input

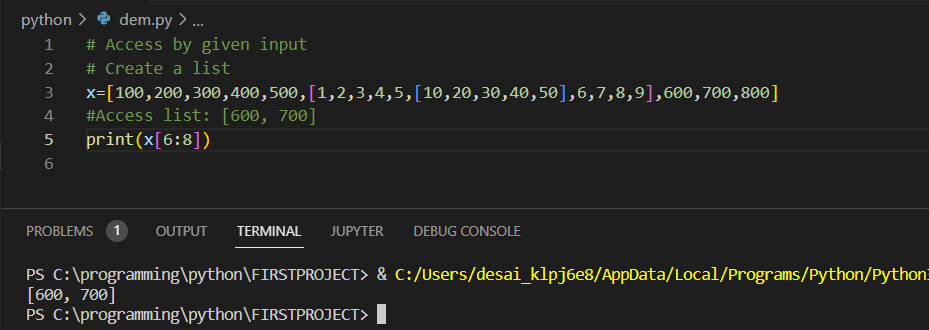
# Create a list

x=[100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]

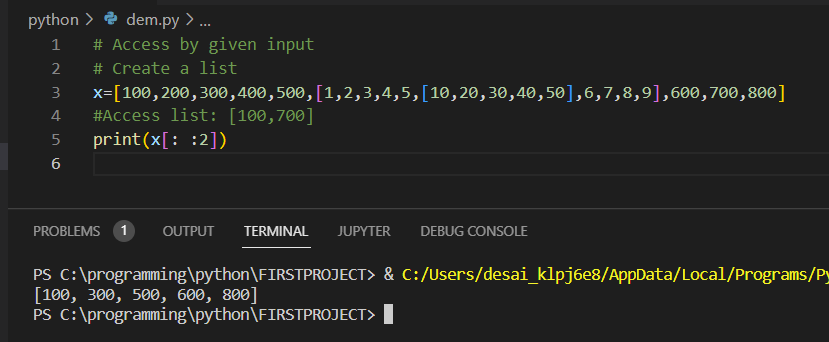
#Access list: [1, 2, 3, 4]

print(x[5][:4])

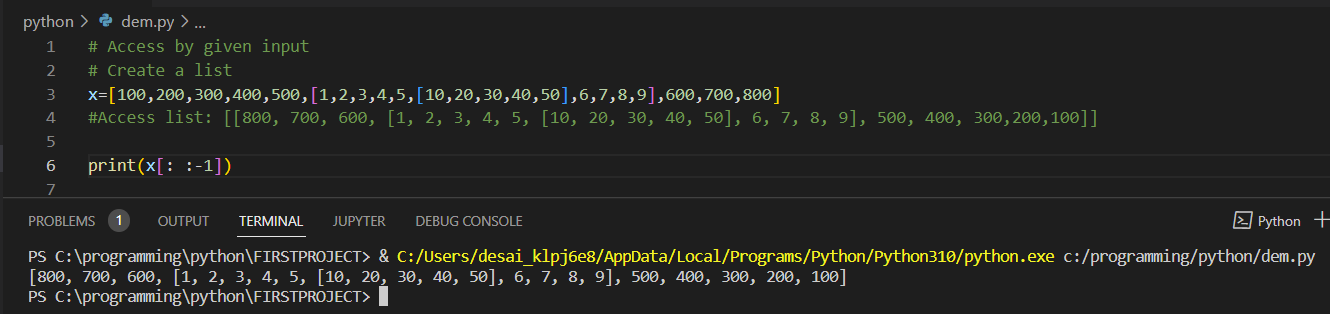
b)Access list: [600, 700]



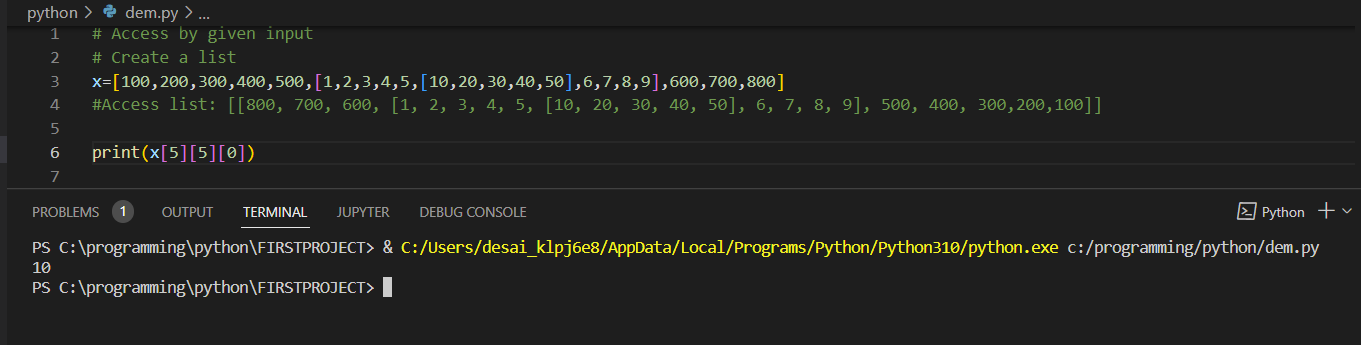
c)Access list: [100, 300, 500, 600, 800]



d) Access list: [[800, 700, 600, [1, 2, 3, 4, 5, [10, 20, 30, 40, 50], 6, 7, 8, 9], 500, 400, 300, 200,100]]



e) Access list:[10]



f) Access list: [ ]

2. Create a list of thousand numbers using range and xrange and see the difference between each other.

Output-



3. How Tuple is beneficial as compared to the list?

Ans-

-Tuples are created faster than lists.

- Tuples are faster because Python allocates memory to tuples in a single and small space, whereas, for lists, the memory is allocated in different chunks linked by pointers.

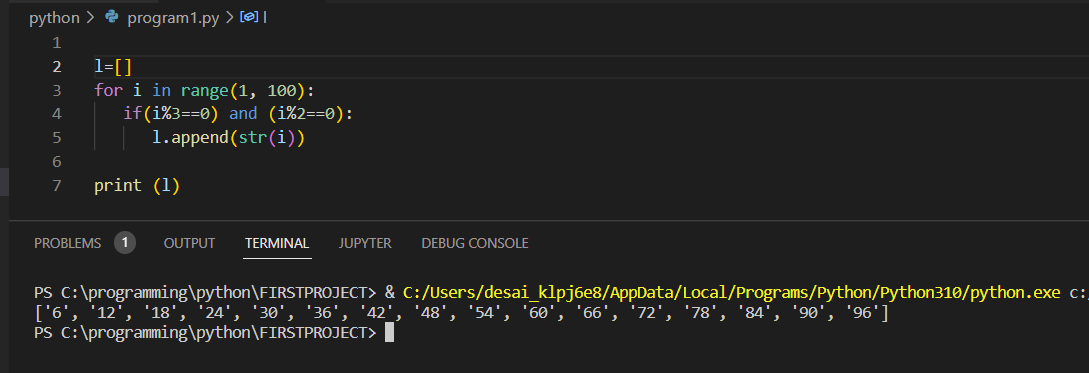
- Memory Consumption by tuples is less as compared to Lists.

- Dictionary uses tuples as key due to their hashable and immutable nature.

- Operations on tuples can be executed faster compared to operations on lists.

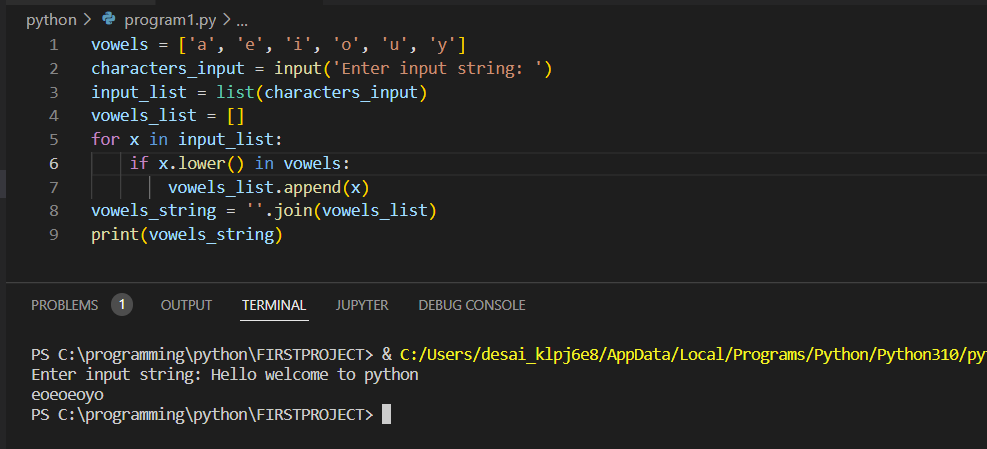
4. Write a program in Python to iterate through the list of numbers in the range of 1,100 and print the number which is divisible by 3 and is a multiple of 2.

Output-



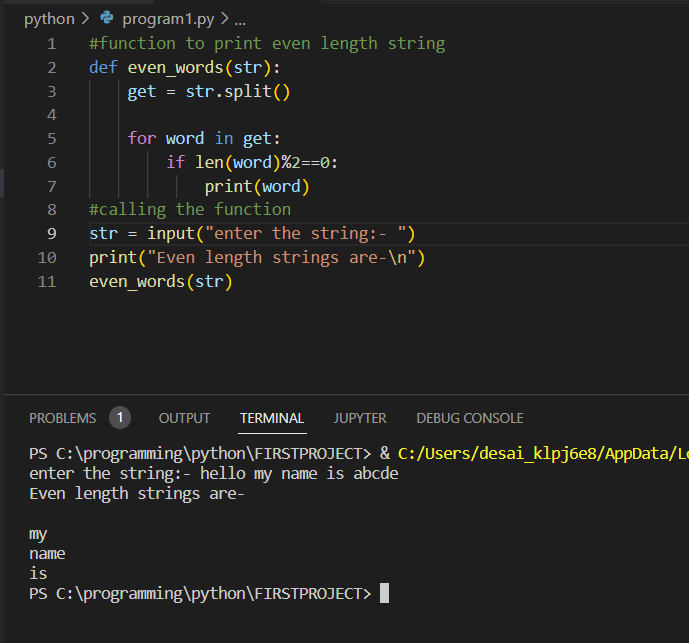
5. Write a program in Python to reverse a string and print only the vowel alphabet if it exists in the string with their index.

Output-



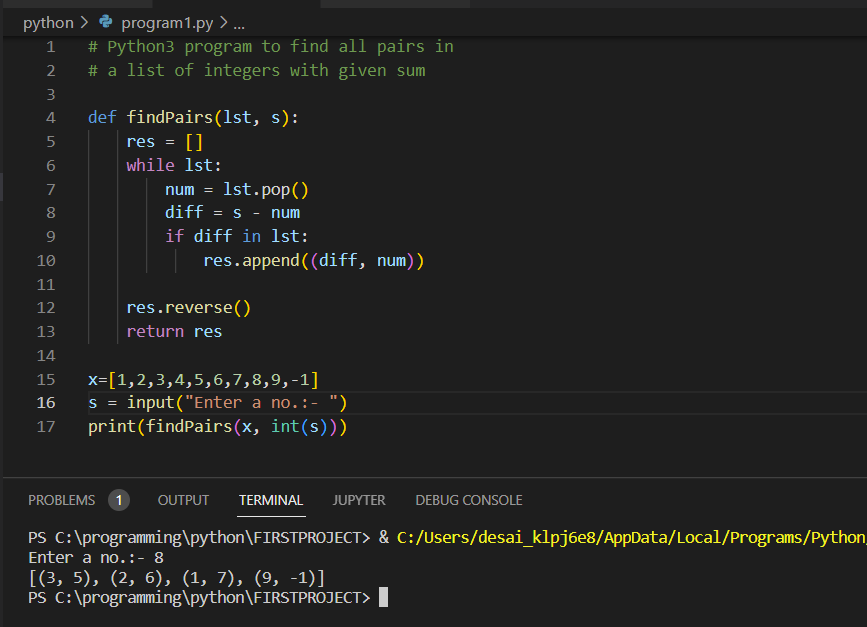
6. Write a program in Python to iterate through the string “hello my name is abcde” and print the string which is having an even length.

Output-



7. Write a program in python to print the pair of numbers whose sum is equal to the result number that is let's say 8. x=[1,2,3,4,5,6,7,8,9,-1]

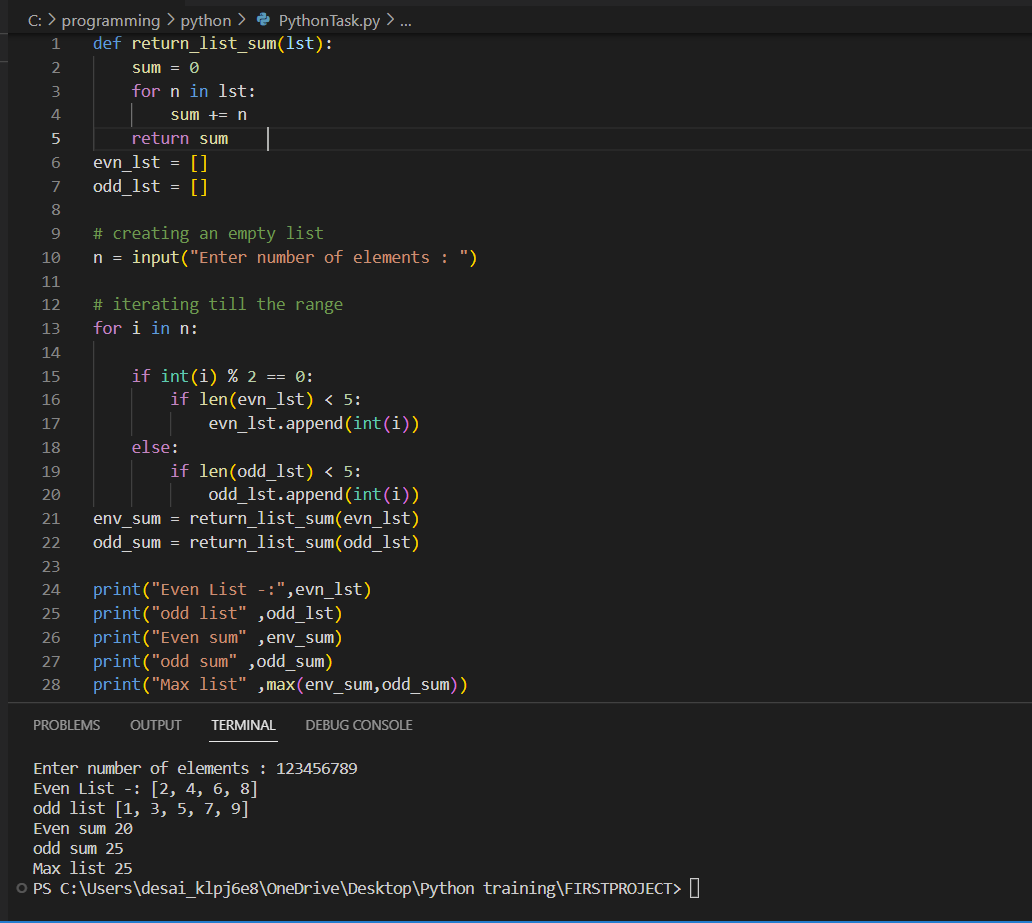
Output-



8. Write a program in Python to complete the following task

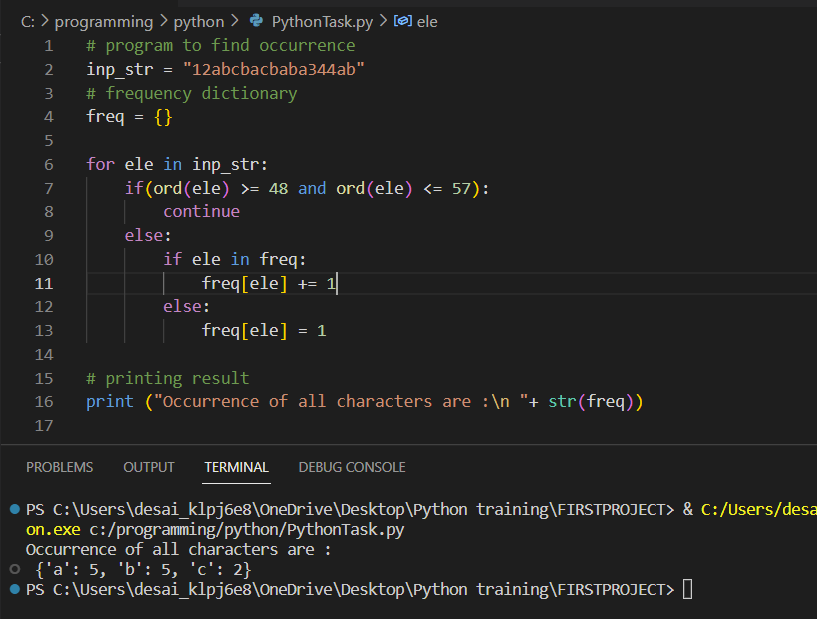
* Create two lists such as even\_list and odd\_list
* Ask user to enter a number in the range of 1,50 and make sure if the entered number is even, append it to the even\_list and if the entered number is odd append it to the odd\_list.
* Keep that in mind you can only add 5 items in each list
* Make sure once you enter all the 5 elements, calculate the sum of the list and return the maximum of the list.

Output-



9. Write a program to find out the occurrence of a specific character from an alphanumeric string. Sample input: 12abcbacbaba344ab Expected output: a=5 b=5 c=2 NOTE: Make sure to avoid counting the occurrence of numeric values in the string.

Output-



10. Generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10)

Output-

