In [18]:

```
"""
1. Write a program that asks the user to enter a list of at least five integers. Do
(a) Print out the total number of items in the list.
(b) Print out the fourth item (index 3) in the list.
(c) Print out the last three items in the list.
(d) Print out all the items in the list except the first two.
(e) Print out the list in reverse order.
(f) Print out the largest and smallest values in the list.
(g) Print out the sum of all the values in the list.
(h) If the list contains a zero, print out the index of the first zero in the list,
(i) Sort the list and print out the list after sorting.
(j) Delete the first item from the (now sorted) list and print out the new list.
(k) Change the second-to-last item in the list to 9876 and print out the new list.
(l) Append the value -500 to the end of the list and print out the new list.

"""

# creating an empty list
lst = []

# number of elements as input
n = int(input("Enter number of elements : "))
# iterating till the range
for i in range(0, n):
 if n<5:
  print("Please enter number >=5 ")
  break
 ele = int(input())
 # adding the element
 lst.append(ele)

# Print out the total number of items in the list.
print("Total number of items in the list :",len(lst))

# Print out the fourth item (index 3) in the list.
print("Fourth item (index 3) in the list :",lst[3])

#Print out the last three items in the list.
print("last three items in the list :",lst[-3:])
#Pint out all the items in the list except the first two.
print("All the items in the list except the first two :",lst[2:])
#Print out the list in reverse order.
lst.reverse()
print("list in reverse order:",lst)
#Print out the largest and smallest values in the list.
print("Largest and smallest values in the list: ",max(lst)," ,",min(lst))
#Print out the sum of all the values in the list.
print("sum of all the values in the list :",sum(lst))

#If the list contains a zero, print out the index of the first zero in the list,
#and otherwise print out a message saying there are no zeroes.
python_indices  = [index for (index, item) in enumerate(lst) if item ==0]
print("indices",python_indices)
if python_indices:
    print(python_indices[0])
else:
    print("no zeros in the list")
#Sort the list and print out the list after sorting.
sorted_lst=sorted(lst)
print("list after sorting",sorted_lst)
#Delete the first item from the (now sorted) list and print out the new list.
sorted_lst.pop(0)
```

```
print('Updated List: ', sorted_lst)
#Change the second-to-last item in the list to 9876 and print out the new list.
sorted_lst[-2]=9876
print("Updated second last item ",sorted_lst)
#Append the value -500 to the end of the list and print out the new list.
sorted_lst.append(-500)
print(sorted_lst)
```

```
Enter number of elements : 5
4
3
8
6
4
Total number of items in the list : 5
Fourth item (index 3) in the list : 6
last three items in the list : [8, 6, 4]
All the items in the list except the first two : [8, 6, 4]
list in reverse order: [4, 6, 8, 3, 4]
Largest and smallest values in the list:  8  , 3
sum of all the values in the list : 25
indices []
no zeros in the list
list after sorting [3, 4, 4, 6, 8]
Updated List:  [4, 4, 6, 8]
Updated second last item  [4, 4, 9876, 8]
[4, 4, 9876, 8, -500]
```

In [22]:

```
"""
Write a program that asks the user to enter a list of numbers. Then print out the
thing in the list and the first index at which it appears in the list.
"""
input_string = input("Enter elements of a list separated by space ")
print("\n")
user_list = input_string.split()
# print list
print("list: ", user_list)
print("smallest element in the list : ", min(user_list))
# get the index of of min element
index = user_list.index(min(user_list))
print("Index of of min element :",index)
```

```
Enter elements of a list separated by space 10 5 6 3 67 1


list:  ['10', '5', '6', '3', '67', '1']
smallest element in the list :  1
Index of of min element : 5
```

In [4]:

```
"""
 Write a program that asks the user to enter a string of lowercase letters and
 creates a list containing counts of how many times each letter appears in the stri
 The first index is how many a’s are in the string, the second is how many b’s, et
"""

from collections import Counter
input_string = input("Enter a string ")
print(input_string)
collection = Counter(input_string)

print(collection)
print(sorted(collection.items()))
```

```
Enter a string welcometoaabbcdef
welcometoaabbcdef
Counter({'e': 3, 'c': 2, 'o': 2, 'a': 2, 'b': 2, 'w': 1, 'l': 1, 'm': 1, 't': 1,
'd': 1, 'f': 1})
[('a', 2), ('b', 2), ('c', 2), ('d', 1), ('e', 3), ('f', 1), ('l', 1), ('m', 1),
('o', 2), ('t', 1), ('w', 1)]
```

In [12]:

```
"""
Create a dictionary whose keys are the strings 'abc', 'def', 'ghi', 'jkl', and 'mno
(a) Print the value in the dictionary associated with the key 'def'.
(b) Use the keys() method to print out all the keys.
(c) Loop over the dictionary and print out all the keys and their associated values
(d) Use an if statement to check if the dictionary contains the key 'pqr' and print
message indicating whether it does or doesn’t.
(e) Change the value associated with the key 'abc' to 23 and then print out all the
dictionary using the values() method.
"""

alphabets_dict = {"abc": 7, "def": 11, "ghi": 13, "jkl": 17, "mno": 19}
# print the value in the dictionary associated with the key 'def'.
print("value in the dictionary associated with the key def:",alphabets_dict["def"]

# Use the keys() method to print out all the keys.
print("keys() method to print out all the keys :",alphabets_dict.keys())

#Loop over the dictionary and print out all the keys and their associated values.
for key in alphabets_dict:
 print(key, '->', alphabets_dict[key])

#Use an if statement to check if the dictionary contains the key 'pqr' and print ou
#message indicating whether it does or doesn’t.
if "pqr" in alphabets_dict:
    print("key Exists")
else:
    print("Key does not exist")
#Change the value associated with the key 'abc' to 23 and then print out all the va
#dictionary using the values() method.

alphabets_dict["abc"] = 23
print("values() method to print out all the vaues :",alphabets_dict.values())
```

```
value in the dictionary associated with the key def: 11
keys() method to print out all the keys : dict_keys(['abc', 'def', 'ghi', 'jkl',
'mno'])
abc -> 7
def -> 11
ghi -> 13
jkl -> 17
mno -> 19
Key does not exist
values() method to print out all the vaues : dict_values([23, 11, 13, 17, 19])
```

In [ ]: