1. Write a Python program to sum all the items in a list.

In [3]:
```
list = [1,2,3,4,5,6,7,8,9,10]
sum(list)
```

Out[3]: 55

2. Write a Python program to get the largest number from a list.

In [4]:
```
max(list)
```

Out[4]: 10

3. Write a Python program to count the number of strings from a given list of strings. The string length is 2 or more and the first and last characters are the same

In [2]:
```
strings_list = ['abc','xyz','aba','1221']

# len of strings_list
strings_list_len = len(strings_list)

# The length of the string is greater than or equal to 2 (len(string) >= 2).
# The first character of the string is the same as the last character of the st
# If both conditions are true for a particular string, the count variable is in
count = 0
for x in strings_list:
    if len(x) >= 2 and x[0] == x[-1]:
        count = count + 1

print(f"the total lenght of string: {strings_list_len} ")
print(f"The string length is 2 or more and the first and last characters are th
```

```
the total lenght of string: 4
The string length is 2 or more and the first and last characters are the sam
e: 2
```

4. Write a Python program to remove duplicates from a list

In [4]:
```
duplicate_list = [2, 4, 10, 20, 5, 2, 20, 4]

# The variable final_list is initialized as an empty list.
# This list will store the unique elements from the input list duplicate
final_list = []

# The function uses a for loop to iterate through each element num in the dupl
# Inside the loop, the if statement checks whether the current element num is
# in the final_list. The condition num not in final_list evaluates to True if

for item in duplicate_list:
    if item not in final_list:
        final_list.append(item)

print(final_list)
```

```
[2, 4, 10, 20, 5]
```

5. Write a Python program to check if a list is empty or not.

In [6]:
```
list1 = [1,2,3,4,5,6,7,8,9,10]

if len(list1) == 0:
    print("empty list")
else:
    print(list1)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

In [8]:
```
list1 = []

if len(list1) == 0:
    print("empty list")
else:
    print(list1)
```

```
empty list
```

6.Write a Python program to filter the list if the length of the character is< 4

In [11]:
```
strings_list = ['abc','xyz','aba','1221']

# The for keyword is used to iterate over each element item in the strings_list
# The item variable takes the value of each element in the list one by one.
# The if statement is used to specify the condition that each element
# (item) should meet in order to be included in the new list.
# The condition in this code is len(item) < 4, which checks whether
# the length of the current item is less than 4 characters.
# If the condition in the if statement is True for a particular item, it will l

[item for item in strings_list if len(item) <4]
```

Out[11]:
```
['abc', 'xyz', 'aba']
```

7.Write a Python program to find the second largest number in a list

In [29]:
```
list4 = [1,3,2,4,5,7,6,8,9,10]

sorted_list = sorted(list4)

print(f"sorted_list : {sorted_list}")

print(f"the second largest number in a list:{sorted_list[-2]}")
```

```
sorted_list : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
the second largest number in a list:9
```

8. Write a Python program to reverse a list at a specific location

In [12]:
```
# Step 1: Create a list named 'list' with some elements
my_list = [1, 3, 2, 4, 5, 7, 6, 8, 9, 10]

# Step 2:  the user to enter a number and store it in the variable 'num'A
num = int(input("Enter a number: "))

# Step 3: Create a new list 'x' containing the first 'num' elements from 'my_l
x = my_list[:num]

# Step 4: Create a new list 'y' containing the elements from 'num' till the en
y = my_list[num:]

# Step 5: Create a new list 'z' that is a reversed version of 'y'
z = y[::-1]

# Step 6: Concatenate 'x' and 'z' to create a new list 'reverse_list_at_a_spec
reverse_list_at_a_specific_location = x + z

# Step 7: Print the individual lists 'x', 'y', 'z', and the final concatenated
print("x:", x)
print("y:", y)
print("z:", z)
print("reverse_list_at_a_specific_location:", reverse_list_at_a_specific_locat
```

```
Enter a number: 6
x: [1, 3, 2, 4, 5, 7]
y: [6, 8, 9, 10]
z: [10, 9, 8, 6]
reverse_list_at_a_specific_location: [1, 3, 2, 4, 5, 7, 10, 9, 8, 6]
```

9. Write a Python program to check if a list is a palindrome or not. Return true otherwise false

In [16]:
```
# initializing list
test_list = [1, 4, 5, 4, 1]

# printing original list
print(f"The original list is{test_list}")

# Reversing the list
reversed_list = test_list[::-1]

# checking if palindrome
if reversed_list == test_list:
    print("true")
else:
    print("false")
```

```
The original list is[1, 4, 5, 4, 1]
true
```

10. Write a Python a program to find the union and intersection of two lists

In [29]:
```
list1 = [1, 2, 3, 4, 5]
list2 = [4, 5, 6, 7, 8]

# Step 1: Concatenate the two lists to create a new list 'connected_list'
connected_list = list1 + list2

# Step 2: Find the union of the elements in 'connected_list' and store it in a
Union_new_list = []
for item in connected_list:
    if item not in Union_new_list:
        Union_new_list.append(item)

# Step 3: Find the intersection of elements between 'list1' and 'list2' and sto
intersection = []
for item in list1:
    if item in list2:
        intersection.append(item)

# Step 4: Print the results of the union and intersection
print(f"Union: {Union_new_list}")
print(f"Intersection: {intersection}")
```

```
Union: [1, 2, 3, 4, 5, 6, 7, 8]
Intersection: [4, 5]
```

11.Write a Python script to sort (ascending and descending) a dictionary by value

In [57]:
```
sample_dict = {'apple': 3, 'banana': 1, 'orange': 2, 'grape': 5, 'kiwi': 4}

# The code uses a dictionary comprehension to create a new dictionary, sorted b
# It iterates through each item (key-value pair) in the original dictionary and
# The lambda function passed to the 'key' parameter specifies that the sorting
# The sorted() function returns a list of tuples with the key-value pairs sorte

{k: v for k, v in sorted(sample_dict.items(), key=lambda item: item[1])}
```

Out[57]:
```
{'banana': 1, 'orange': 2, 'apple': 3, 'kiwi': 4, 'grape': 5}
```

12. Write a Python script to check whether a given key already exists in a dictionary.

In [ ]:
```
# The input dictionary containing fruits as keys and their corresponding quant
sample_dict = {'apple': 3, 'banana': 1, 'orange': 2, 'grape': 5, 'kiwi': 4}

# The 'input' function  the user to enter a key name and stores the entered va
a = input("Enter the key name: ")

# The 'if' statement checks if the entered key ('a') exists in the 'sample_dic
if a in sample_dict:
    # If the key exists in the dictionary, it prints a message indicating that
    print(f"{a} key already exists in the dictionary.")
else:
    # If the key does not exist in the dictionary, it prints a message indicat
    print("Key does not exist in the dictionary.")
```

13.Write a Python program to sum all the values in a dictionary

In [2]:
```
# The input dictionary containing fruits as keys and their corresponding quant
sample_dict = {'apple': 3, 'banana': 1, 'orange': 2, 'grape': 5, 'kiwi': 4}

# Create an empty list to store the values from the dictionary.
m_t_list = []

# Iterate through each key in the dictionary using a for loop.
for item in sample_dict:
    # Append the value associated with the current key ('item') to the 'm_t_lis
    m_t_list.append(sample_dict[item])

    # Calculate the sum of all values in the 'm_t_list' and store it in the 'f
    # Note: The 'final' variable will be updated in each iteration, but it wil
    final = sum(m_t_list)

# After the loop completes, the 'final' variable will contain the sum of all v

# Print the result.
print(f"sum of all values in the dictionary {final}")
```

```
sum of all values in the dictionary 15
```

14.Write a Python program to create a dictionary with a number and its corresponding square from 1 to input number. And also check if the input number is less than 10

In [2]:
```
# Create an empty dictionary.
a_dict = {}

# Get an integer input from the user and store it in the 'input_number' variab
input_number = int(input("enter a number: "))

# Check if the entered number is less than or equal to 10.
if input_number <= 10:
    # If the number is less than or equal to 10, create a key-value pair in the
    # The input number is the key, and its square is the value.
    a_dict[input_number] = input_number * input_number

    # Print the dictionary with the newly added key-value pair.
    print(a_dict)
else:
    # If the number is greater than 10, print a message indicating that it shou
    print("the number should be less than 10 ")
```

```
enter a number: 3
{3: 9}
```

15.Write a Python program to sort a given dictionary by key

In [6]:
```
sample_dict = {'apple': 3, 'banana': 1, 'orange': 2, 'grape': 5, 'kiwi': 4}

sorted(sample_dict)
```

Out[6]:
```
['apple', 'banana', 'grape', 'kiwi', 'orange']
```

16. Write a Python program to create a dictionary from a string. Note: Track the count of the letters from the string.

In [8]:
```
# Get a string input from the user and store it in the 'input_string' variable
input_string = input("enter a string: ")

# Create an empty dictionary to store the letter counts.
letter_count = {}

# Iterate through each character in the 'input_string' using a for loop.
for x in input_string:
    # Check if the current character 'x' is an alphabet letter using the 'isalp
    if x.isalpha():
        # Convert the character to lowercase using the 'lower()' method.
        x_lower = x.lower()

        # Use the 'get' method to retrieve the count of the current letter fro
        # If the letter is not present in the dictionary, the 'get' method retu
        # Then, increment the count of the current letter by 1.
        letter_count[x_lower] = letter_count.get(x_lower, 0) + 1

# After the loop completes, the 'letter_count' dictionary will contain the cou

# Print the resulting 'letter_count' dictionary.
print(letter_count)
```

```
enter a string: anu rashik
{'a': 2, 'n': 1, 'u': 1, 'r': 1, 's': 1, 'h': 1, 'i': 1, 'k': 1}
```

17. Write a Python program to get the top three items in a shop

In [ ]:
```
# The input dictionary containing items as keys and their corresponding prices
mydict = {'item1': 45.50, 'item2': 35, 'item3': 41.30, 'item4': 55, 'item5': 2

# Extract the values from the dictionary and store them in the 'resultList' var
resultList = list(mydict.values())

# Sort the 'resultList' in ascending order using the 'sorted' function and stor
x = sorted(resultList)

# Print the 'x' list starting from the third element (index 2) until the end.
print(x[2:])
```