In [63]:
```
'''QUESTION ONE
Write a program that finds the maximum value of the given list, assuming that the list
```

Out[63]:
```
'QUESTION ONE\nWrite a program that finds the maximum value of the given list, assumi
ng that the list contains at least one element.'
```

In [65]:
```
def find_max_value(input_list):
    # Assuming the list is not empty
    max_value = max(input_list)
    return max_value

# Example usage:
my_list = [2,4,7,4,23,5,1,4,8,9]
max_value = find_max_value(my_list)
print("Maximum value:", max_value)
```

```
Maximum value: 23
```

In [66]:
```
'''QUESTION TWO

. Write a program that calculates the average value of the given list.

SOLUTION

'''
```

Out[66]:
```
'QUESTION TWO\n\n. Write a program that calculates the average value of the given lis
t.\n\nSOLUTION\n\n'
```

In [68]:
```
def calculate_average(input_list):
    # Assuming the list is not empty
    sum_values = sum(input_list)
    average = sum_values / len(input_list)
    return average

# Example usage:
my_list = [4,7,1,5,11,53,12,46,84,23]
average_value = calculate_average(my_list)
print("Average value:", average_value)
```

```
Average value: 24.6
```

In [ ]:
```
'''QUESTION THREE

Write a program that prints the given list of integers in reverse order.

'''
```

In [70]:
```
def print_reverse_order(input_list):
    input_list.reverse()
    print("List in reverse order:", input_list)

# Example usage:
my_list = [2,6,7,45,23,53,14,45,89,5]
print_reverse_order(my_list)
```

```
List in reverse order: [5, 89, 45, 14, 53, 23, 45, 7, 6, 2]
```

In [ ]:
```
'''
    QUESTION FOUR
Write a program that accepts two lists of integers and prints true
if each element in the first list is less than the element at the same index in the sec
Your program should print false if the lists are not the same length.'''
```

In [73]:
```
def compare_lists(list1, list2): # accepting the two lists
    # Check if the lists have the same length
    if len(list1) != len(list2):
        print("false")
        return

    # Compare elements at each index
    for i in range(len(list1)):
        if list1[i] >= list2[i]:
            print("false")
            return

    # If the loop completes without returning, all elements are less
    print("true")

# Example usage:
list1 = [1, 2, 3, 4]
list2 = [5, 6, 7, 8]

compare_lists(list1, list2)
```

```
true
```

In [ ]:
```

'''
QUESTION FIVE
Write a program that accepts a list of integers and two indexes and swaps the elements
'''
```

In [34]:
```
def swap_elements(my_list, index1, index2):
    # Check if the indexes are within the valid range
    if 0 <= index1 < len(my_list) and 0 <= index2 < len(my_list):
        # Swap the elements at the given indexes
        my_list[index1], my_list[index2] = my_list[index2], my_list[index1]
        print("List after swapping:", my_list)
    else:
        print("Invalid indexes")

# Example usage:
input_list = [5, 2, 3, 4, 5]
index_to_swap1 = 1
index_to_swap2 = 10

print("Original list:", input_list)
swap_elements(input_list, index_to_swap1, index_to_swap2)
```

```
Original list: [5, 2, 3, 4, 5]
Invalid indexes
```

In [ ]:
```
'''
QUETION SIX
Write a program that accepts two lists of integers and prints a new list containing all
elements of the first list followed by all elements of the second.'''
```

In [74]:
```
def acceptLists(list1,list2):
    list1.extend(list2)
    print(list1)
    return

list1 = [1, 2, 3, 4]
list2 = [5, 6, 7, 8]

acceptLists(list1,list2)
```

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

In [ ]:
```
'''
QUESTION SEVEN
Write a program that accepts a list of integers and an integer value as its parameters
prints the last index at which the value occurs in the list. The program should print -
For example, in the list [74, 85, 102, 99, 101, 85, 56], the last index of the value 85
```

In [76]:
```
def find_last_index(my_list, value):
    # Start from the end of the list and iterate backwards
    for i in range(len(my_list) - 1, -1, -1):
        if my_list[i] == value:
            print("The last index of", value, "is:", i)
            return

    # If the value is not found
    print(value, "not found in the list. Last index is -1.")

# Example usage:
my_list = [74, 85, 102, 99, 101, 85, 56]
search_value = 85
find_last_index(my_list, search_value)
```

```
The last index of 85 is: 5
```

In [ ]:

In [ ]:
```
'''
QUESTION EIGHT
Write a program that prints the range of values in a list of integers.
The range is defined as 1 more than the difference between the maximum and
minimum values in the list. For example,

if a list contains the values [36, 12, 25, 19, 46, 31, 22],
the program should return 35. You may assume that the list has at least one element.''
```

In [48]:
```
def rangeValue(mylist):
    max_value=max(mylist)
    min_value=min(mylist)
    range_value=(max_value-min_value) + 1
    print(range_value)
    return
myList =[36, 12, 25, 19, 46, 31, 22]
rangeValue(myList)
```

```
35
```

In [ ]:
```
'''
QUESTION NINE
Write a program that accepts a list of integers, a minimum value, and a maximum value a
the count of how many elements from the list fall between the minimum and maximum (incl

For example, in the list [14, 1, 22, 17, 36, 7, -43, 5], for minimum value 4 and maximu
there are four elements whose values fall between 4 and 17.'''
```

In [ ]:

In [78]:
```
def count_elements_in_range(my_list, min_value, max_value):
    # Initialize a counter for elements in the range
    count = 0
    print("initial count" ,count)

    # Iterate through the list
    for i in my_list:
        # Check if the element is within the specified range
        if min_value <= i <= max_value:
            count =count +1
            print("my_list", i ,"count value",count)

    # Print the count of elements in the range
    print("Count of elements between", min_value, "and", max_value, "is:", count)

# Example usage:
my_list = [14, 1, 22, 17, 36, 7, -43, 5]
min_value = 4
max_value = 17

count_elements_in_range(my_list, min_value, max_value)
```

```
initial count 0
my_list 14 count value 1
my_list 17 count value 2
my_list 7 count value 3
my_list 5 count value 4
Count of elements between 4 and 17 is: 4
```

In [ ]:

In [ ]:
```
'''
QUESTION TEN
Write a program that accepts a list of real numbers and prints true if the list is in s
order and false otherwise.

For example, if lists named list1 and list2 store [16.1, 12.3, 22.2, 14.4]
and [1.5, 4.3, 7.0, 19.5, 25.1, 46.2] respectively, the program should print false for

Assume the list has at least one element. A one-element list is sorted.

'''
```

In [ ]:

In [79]:
```
def is_sorted(my_list):
    # Create a sorted copy of the list
    sorted_list = sorted(my_list)
    print(sorted_list)

    # Check if the original list is the same as the sorted list
    return my_list == sorted_list

# Example usage:
list1 = [16.1, 12.3, 22.2, 14.4]
list2 = [1.5, 4.3, 7.0, 19.5, 25.1, 46.2]

print("Is list1 sorted?", is_sorted(list1))
print("Is list2 sorted?", is_sorted(list2))
```

```
[12.3, 14.4, 16.1, 22.2]
Is list1 sorted? False
[1.5, 4.3, 7.0, 19.5, 25.1, 46.2]
Is list2 sorted? True
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: