365 DataScience Binary search in Python

Step 1 Create a function that performs a binary search on a list

```
# Binary search is an algorithm that relies upon the fact that the list
 is sorted.

def binary_search(item, my_list):
    # Make sure that the list is sorted.
    my_list_sorted = sorted(my_list)
    # Create a boolean variable which initially assumes that the item i
s not found.
    found = False
    # Keep track of the index of the first item in the list
    first = 0
    # Keep track of the index of the last item in the list
    last = len(my_list_sorted) - 1

    # While the index of the first item is less than or equal to the in
dex of the last item
    # and the item is still not found, perform the operations in the bo
dy of the loop.
    while first <= last and found == False:

        # Calculate the index of the item that sits in the middle of th
e list.
        # We use floor division as we want integers and not decimal num
bers.
        midpoint = (first + last)//2

        # Check if this middle item is the item we are looking for.
        if my_list_sorted[midpoint] == item:

            # If it is, set the boolean variable to True
            found = True

        # If it is not, do the followng:
        else:

            # Check if the value of the midpoint is less than the item
we are looking for
            if my_list_sorted[midpoint] < item:

                # Increase the initial index by one
                first = midpoint + 1

            # If the value of the midpoint is greater than the item we
```

```
are looking for
            else:

                # Decrease the last index by one
                last = midpoint - 1

    # Return whether we have found the item or not.
    return found
```

Step 2 Check if the function performs as expected

```
# Create a list of numbers
test = [6, 5, 8, 2, 3, 45, 87, 24, 70]

# Check if the number 4 is inside this list.
print(binary_search(4,test))

# Check if the number 87 is inside this list.
print(binary_search(87,test))
```