```python
"""
CS241 Team Activity 10 - Merge
Written by Chad Macbeth
"""

"""
File: ta10-solution.py
Author: Br. Burton
This file demonstrates the merge sort algorithm. There
are efficiencies that could be added, but this approach is
made to demonstrate clarity.
"""

from random import randint
MAX_NUM = 100


def merge_sort(items):
    """
    Sorts the items in the list
    :param items: The list to sort
    """

    # If there is only one item, then its already sorted (base case)
    if len(items) <= 1:
        return

    # Create the 2 halves.  At first they are unsorted
    middle = len(items) // 2
    left = items[:middle]  # 0 <= i < middle
    right = items[middle:] # middle <= i <= end

    # Sort both sides (recursive calls)
    merge_sort(left)
    merge_sort(right)

    # Perform the merge now that both sides have been sorted
    left_side_pos = 0
    right_side_pos = 0
    merged_pos = 0

    # Select the smallest from both sides and create a sorted list
    # If we run out of numbers from one of the lists, then exit
    # the while loop
    while left_side_pos < len(left) and right_side_pos < len(right):
        # Left side has the next smallest number ... use it
        if left[left_side_pos] < right[right_side_pos]:
            items[merged_pos] = left[left_side_pos]
            left_side_pos += 1
        # Right side has the next smallest number ... use it
        else:
            items[merged_pos] = right[right_side_pos]
            right_side_pos += 1

        merged_pos += 1

    # If there are still numbers on the left side, just copy them over
    # since they are already sorted
    while left_side_pos < len(left):
        items[merged_pos] = left[left_side_pos]
        left_side_pos += 1
        merged_pos += 1

    # If there are still numbers on the right side, just copy them over
    # since they are already sorted
    while right_side_pos < len(right):
        items[merged_pos] = right[right_side_pos]
        right_side_pos += 1
        merged_pos += 1


def generate_list(size):
    """
    Generates a list of random numbers.
    """
    items = [randint(0, MAX_NUM) for i in range(size)]
    return items


def display_list(items):
    """
    Displays a list
    """
    for item in items:
        print(item)


def main():
    """
    Tests the merge sort
    """
    size = int(input("Enter size: "))

    items = generate_list(size)
    merge_sort(items)

    print("\nThe Sorted list is:")
    display_list(items)


if __name__ == "__main__":
    main()

```