202) Given a list of item weights and the maximum capacity of a container, determine the maximum weight that can be loaded into the container using a greedy approach. The greedy approach should prioritize loading heavier items first until the container reaches its capacity.

Test Case 1:
Input:
n = 5
weights = [10, 20, 30, 40, 50]
max_capacity = 60
Output: 50
Test Case 2:
Input:
n = 6
weights = [5, 10, 15, 20, 25, 30]
max_capacity = 50
Output: 50

AIM: To write a python program for the greedy approach should prioritize loading heavier items first until the container reaches its capacity.
PROGRAM:

```
def max_weight_greedy(weights, max_capacity):
    # Sort weights in descending order
    weights.sort(reverse=True)

    current_weight = 0

    for weight in weights:
        if current_weight + weight <= max_capacity:
            current_weight += weight
        else:
            break

    return current_weight

n1 = 5
weights1 = [10, 20, 30, 40, 50]
max_capacity1 = 60
print(max_weight_greedy(weights1, max_capacity1))
```

OUTPUT:

TIME COMPLEXITY : O(n logn)