TOPIC 5.9 : GREEDY KNAPSACK FOR MAXIMUM WEIGHT LOADING

Problem Statement  
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 program that selects items to load into a container such that the total weight is maximized without exceeding the maximum capacity, using a greedy approach that picks heavier items first.

Algorithm

1. Start
2. Sort the list of weights in descending order
3. Initialize total = 0
4. For each weight in the sorted list:
   * If total + weight ≤ max\_capacity, add weight to total
   * Otherwise, skip
5. Return total
6. Stop

Input and Output  
A screenshot of a computer

AI-generated content may be incorrect.

Result  
The program successfully computes the maximum weight that can be loaded into the container using the greedy method.

Performance Analysis  
Time Complexity: O(n log n) due to sorting  
Space Complexity: O(1) for variables