37 Container loader

#include <stdio.h>

int container\_loading\_dp(int containers[], int n, int capacity) {

// Create a 2D table to store dp values

int dp[n + 1][capacity + 1];

// Initialize base cases

for (int i = 0; i < n + 1; i++) {

dp[i][0] = 0; // 0 containers needed for 0 capacity

}

for (int j = 0; j < capacity + 1; j++) {

dp[0][j] = INT\_MAX; // Maximum possible value (infinity)

}

// Fill the dp table using the logic

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= capacity; j++) {

// Exclude the current container

dp[i][j] = dp[i - 1][j];

// Include the current container if it fits

if (containers[i - 1] <= j) {

dp[i][j] = min(dp[i][j], 1 + dp[i - 1][j - containers[i - 1]]);

}

}

}

// Return the minimum number of containers or -1 if not possible

return dp[n][capacity] != INT\_MAX ? dp[n][capacity] : -1;

}