/\* Coin Changing Problem \*/

#include <iostream>

#include <vector>

#include <climits> // For INT\_MAX

using namespace std;

int minCoins(int A, const vector<int>& coins) {

// Initialize dp array with maximum value

vector<int> dp(A + 1, INT\_MAX);

// Base case: 0 coins needed to make up 0 amount

dp[0] = 0;

// Iterate over each denomination of coins

for (int i = 0; i < coins.size(); ++i) {

// Iterate over each amount

for (int j = coins[i]; j <= A; ++j) {

// Update dp[j] to the minimum of its current value and dp[j - coins[i]] + 1

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

}

}

// Return the minimum number of coins required to make up amount A

return dp[A];

}

int main() {

// Input the number of denominations

int n;

cout << "Enter the number of denominations: ";

cin >> n;

// Input the denominations of coins

vector<int> coins(n);

cout << "Enter the denominations of coins: ";

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

cin >> coins[i];

}

// Input the target amount

int A;

cout << "Enter the target amount: ";

cin >> A;

// Calculate the minimum number of coins required

int minCoinsRequired = minCoins(A, coins);

cout << "Minimum number of coins required to make up amount " << A << ": " << minCoinsRequired << endl;

return 0;

}