Maximizing Club Memberships

Time limit per test: 1 Seconds.

Memory limit per test: 256 Megabytes.

Problem Statement:

At NIT Agartala, students are enthusiastic about joining various clubs and associations offered by the college. Each club has a specific membership fee, and students want to maximize their participation across different clubs by utilizing their available funds optimally.

Given the budget a student has and the membership fees of each club, which can only be one of `f1`, `f2`, or `f3`, determine the maximum number of clubs they can join without exceeding their budget.

Input

The input consists of a single line containing four space-separated integers `budget`, `f1`, `f2`, and `f3` (1 ≤ budget, f1, f2, f3 ≤ 4000), where:

- `budget` is the amount of money a student has.
- `f1`, `f2`, and `f3` are the membership fees of three different clubs.

Output

Print a single integer — the maximum number of clubs the student can join without exceeding their budget.

Examples

Input:

5 5 3 2

Output:

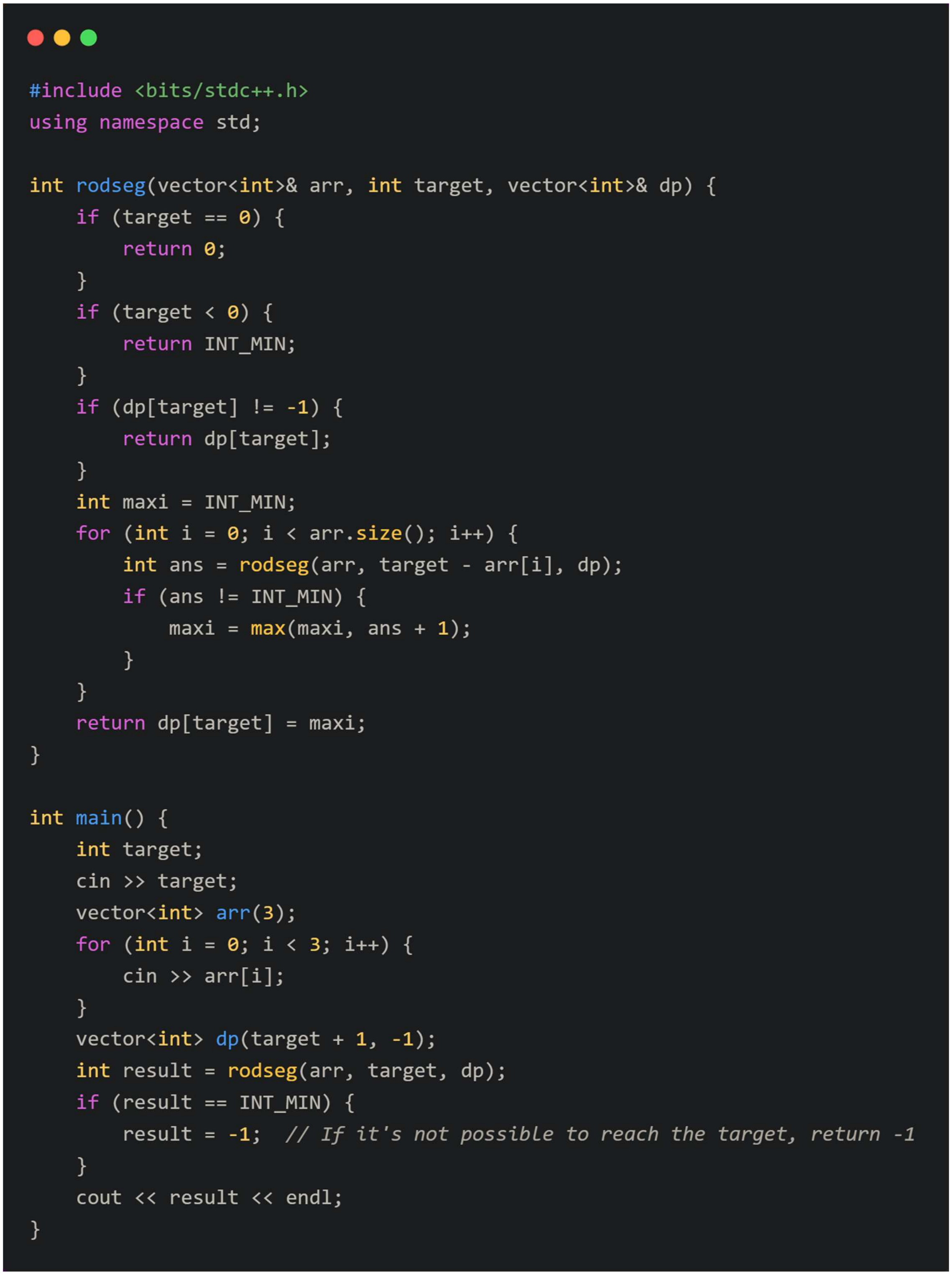
2

Input:

7 5 5 2

Output:

2



```cpp
#include <bits/stdc++.h>
using namespace std;

int rodseg(vector<int>& arr, int target, vector<int>& dp) {
    if (target == 0) {
        return 0;
    }
    if (target < 0) {
        return INT_MIN;
    }
    if (dp[target] != -1) {
        return dp[target];
    }
    int maxi = INT_MIN;
    for (int i = 0; i < arr.size(); i++) {
        int ans = rodseg(arr, target - arr[i], dp);
        if (ans != INT_MIN) {
            maxi = max(maxi, ans + 1);
        }
    }
    return dp[target] = maxi;
}

int main() {
    int target;
    cin >> target;
    vector<int> arr(3);
    for (int i = 0; i < 3; i++) {
        cin >> arr[i];
    }
    vector<int> dp(target + 1, -1);
    int result = rodseg(arr, target, dp);
    if (result == INT_MIN) {
        result = -1;  // If it's not possible to reach the target, return -1
    }
    cout << result << endl;
}
```