Birthday Chocolate ☆

47.53 more points to get your gold badge!

Rank: 143605 | Points: 802.47/850

Problem Submissions Leaderboard Editorial

Editorial by StefanK

Approach

We're going to solve this problem using a brute force approach by looping through the squares in the chocolate bar at each possible interval of squares that can form a piece of length m and summing the values of the squares within that interval. If that sum is equal to d, we increment the integer that stores the number of ways the chocolate can be shared with Ron. Once we finish traversing through all possible intervals, we print the number of ways we can satisfy the constraints as output.

Tip: Be sure to set up your loop to prevent "array index out of bounds" errors!

```
#include <bits/stdc++.h>

using namespace std;

int main(){
    int n;
    cin >> n;
    vector<int> c(n);
    for(int i = 0; i < n; i++){
       cin >> c[i];
    }
    int d;
    int m;
    cin >> d >> m;

    int ans = 0;
    for(int i = 0; i < n; i++){
        if (i + m - 1 < n){
            int sum = 0;
            for (int j = i; j <= i + m - 1; j++){
                sum = sum + c[j];
            }
            if (sum==d) {
                ans++;
            }
        }
    }
    cout << ans << endl;
    return 0;
}
```

We can also solve this challenge in $\mathcal{O}(n)$ time (see Tester solution below).

Tested by AllisonP

Problem Tester's code:

Java

```
import java.util.*;

public class Solution {

    public static int getNumberOfWays(int n, int d, int m, int[] sum) {
        // Modify array to make each 'i' contain the running sum of prior elements
        for (int i = 1; i < n; i++) {
            sum[i] += sum[i - 1];
        }

        // Set number of ways counter
        // If there are >= 'm' squares AND the first possible piece has sum = 'd', 1
        // Else, 0
        int numberOfWays = (m <= n && sum[m - 1] == d) ? 1 : 0;

        // Check the sums for pieces ending at index 'm' through 'n - 1'
        for (int i = m; i < n; i++) {
            // If the sum of the piece is equal to 'd'
            if (sum[i] - sum[i - m] == d) {
                // Increment ways counter
                numberOfWays++;
            }
        }

        return numberOfWays;
    }

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int[] squares = new int[n];
        for(int squares_i=0; squares_i < n; squares_i++){
            squares[squares_i] = in.nextInt();
        }
        int d = in.nextInt();
        int m = in.nextInt();
        in.close();

        System.out.println(getNumberOfWays(n, d, m, squares));
    }
}
```

Feedback

Was this editorial helpful?

Yes No

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature