

Subarray Division 1

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

using namespace std;

string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);

/*
 * Complete the 'birthday' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts following parameters:
 * 1. INTEGER_ARRAY s
 * 2. INTEGER d
 * 3. INTEGER m
 */

int birthday(vector<int> s, int d, int m) {

    int count=0;
    int n=s.size();
    for(int i=0;i<=n-m;i++){
        int sum=0;
        for(int j=i;j<m+i;j++){
            sum+=s[j];
        }
        if(sum==d){
            count++;
        }
    }

    return count;
}

int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));

    string n_temp;
    getline(cin, n_temp);

    int n = stoi(ltrim(rtrim(n_temp)));
```

```

string s_temp_temp;
getline(cin, s_temp_temp);

vector<string> s_temp = split(rtrim(s_temp_temp));

vector<int> s(n);

for (int i = 0; i < n; i++) {
    int s_item = stoi(s_temp[i]);

    s[i] = s_item;
}

string first_multiple_input_temp;
getline(cin, first_multiple_input_temp);

vector<string> first_multiple_input =
split(rtrim(first_multiple_input_temp));

int d = stoi(first_multiple_input[0]);

int m = stoi(first_multiple_input[1]);

int result = birthday(s, d, m);

fout << result << "\n";

fout.close();

return 0;
}

string ltrim(const string &str) {
    string s(str);

    s.erase(
        s.begin(),
        find_if(s.begin(), s.end(), not1(ptr_fun<int,
int>(isspace)))
    );

    return s;
}

string rtrim(const string &str) {

```

```

    string s(str);

    s.erase(
        find_if(s.rbegin(), s.rend(), not1(ptr_fun<int,
int>(isspace))).base(),
        s.end()
    );

    return s;
}

vector<string> split(const string &str) {
    vector<string> tokens;

    string::size_type start = 0;
    string::size_type end = 0;

    while ((end = str.find(" ", start)) != string::npos) {
        tokens.push_back(str.substr(start, end - start));

        start = end + 1;
    }

    tokens.push_back(str.substr(start));

    return tokens;
}

```