

All Contests > Binary Battles Round 3 > Beautiful Triplets

Beautiful Triplets

[Problem](#)

[Submissions](#)

[Leaderboard](#)

[Discussions](#)

Given a sequence of integers a , a triplet $(a[i], a[j], a[k])$ is beautiful if:

- $i < j < k$
- $a[j] - a[i] = a[k] - a[j] = d$

Given an increasing sequence of integers and the value of d , count the number of beautiful triplets in the sequence.

Example

$arr = [2, 2, 3, 4, 5]$

$d = 1$

There are three beautiful triplets, by index: $[i, j, k] = [0, 2, 3], [1, 2, 3], [2, 3, 4]$. To test the first triplet, $arr[j] - arr[i] = 3 - 2 = 1$ and $arr[k] - arr[j] = 4 - 3 = 1$.

Function Description

Complete the *beautifulTriplets* function in the editor below.

beautifulTriplets has the following parameters:

- *int d*: the value to match
- *int arr[n]*: the sequence, sorted ascending

Returns

- *int*: the number of beautiful triplets

Input Format

The first line contains **2** space-separated integers, n and d , the length of the sequence and the beautiful difference. The second line contains n space-separated integers $arr[i]$.

Constraints

- $1 \leq n \leq 10^4$
- $1 \leq d \leq 20$
- $0 \leq arr[i] \leq 2 \times 10^4$
- $arr[i] > arr[i - 1]$

Sample Input

STDIN	Function
-----	-----

```
7 3           arr[] size n = 7, d = 3
1 2 4 5 7 8 10 arr = [1, 2, 4, 5, 7, 8, 10]
```

Sample Output

3

Explanation

There are many possible triplets ($\text{arr}[i], \text{arr}[j], \text{arr}[k]$), but our only beautiful triplets are $(1, 4, 7)$, $(4, 7, 10)$ and $(2, 5, 8)$ by value, not index. Please see the equations below:

$$7 - 4 = 4 - 1 = 3 = d$$

$$10 - 7 = 7 - 4 = 3 = d$$

$$8 - 5 = 5 - 2 = 3 = d$$

Recall that a beautiful triplet satisfies the following equivalence relation: $\text{arr}[j] - \text{arr}[i] = \text{arr}[k] - \text{arr}[j] = d$ where $i < j < k$.



The contest has not yet started. It begins in an hour.

Submissions: 0

Max Score: 0

Difficulty: Easy

Rate This Challenge:



[More](#)



```

1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 string ltrim(const string &);
6 string rtrim(const string &);
7 vector<string> split(const string &);
8
9 /*
10  * Complete the 'beautifulTriplets' function below.
11  *
12  * The function is expected to return an INTEGER.
13  * The function accepts following parameters:
14  * 1. INTEGER d
15  * 2. INTEGER_ARRAY arr
16  */
17
18 int beautifulTriplets(int d, vector<int> arr) {
19
20 }
21
22 int main()
23 {
24     ofstream fout(getenv("OUTPUT_PATH"));
25
26     string first_multiple_input_temp;
27     getline(cin, first_multiple_input_temp);
28
29     vector<string> first_multiple_input = split(rtrim(first_multiple_input_temp));
30 }
```

```
31▼ int n = stoi(first_multiple_input[0]);
32
33▼ int d = stoi(first_multiple_input[1]);
34
35 string arr_temp_temp;
36 getline(cin, arr_temp_temp);
37
38 vector<string> arr_temp = split(rtrim(arr_temp_temp));
39
40 vector<int> arr(n);
41
42▼ for (int i = 0; i < n; i++) {
43▼     int arr_item = stoi(arr_temp[i]);
44
45▼     arr[i] = arr_item;
46 }
47
48 int result = beautifulTriplets(d, arr);
49
50 fout << result << "\n";
51
52 fout.close();
53
54 return 0;
55 }
56
57▼ string ltrim(const string &str) {
58     string s(str);
59
60     s.erase(
61         s.begin(),
62         find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
63     );
64
65     return s;
66 }
67
68▼ string rtrim(const string &str) {
69     string s(str);
70
71     s.erase(
72         find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
73         s.end()
74     );
75
76     return s;
77 }
78
79▼ vector<string> split(const string &str) {
80     vector<string> tokens;
81
82     string::size_type start = 0;
83     string::size_type end = 0;
84
85▼ while ((end = str.find(" ", start)) != string::npos) {
86         tokens.push_back(str.substr(start, end - start));
87
88         start = end + 1;
89     }
90
91     tokens.push_back(str.substr(start));
92
93     return tokens;
94 }
```

[Upload Code as File](#) Test against custom input[Run Code](#)[Submit Code](#)[Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |