Magic Potion

Attempted by: **249** / Accuracy: **63%** / Maximum Score: **30** / 3 Votes

Tag(s): Algorithms, Binary Search, Easy-Medium, Searching Edit

PROBLEM | EDITORIAL | MY SUBMISSIONS

Shruti loves to play with Array. She would always be busy doing some random operation with her array. Today she got to know about **Magic Potion**. A Magic Potion is a special power that allows you to remove one element from your array either from the start or the end. After spending some time on Magic Potion she decided to use it on her arrays.

Shruti has an array of size **N**. She calls an array a **Good** array if the sum of the array is exactly **K**. She wants to apply Magic Potion on her array so that she could get a Good array. She wants to find the count of all the Good arrays that could be formed from the given initial array by applying Magic Potion on them any number of times. She is also interested in finding out the minimum Magic Potion required to form a single Good array. Since she is already quite busy she asks you for help.

Input:

The first line contains 2 integers **N** and **K** denoting the size of the array and the sum value required for an array to be Good.This is followed by N-space separated integers that denote the array.

Output:

Print two space-separated integers where the first integers denote the count of all possible Good arrays while the second integers indicate minimum Magic Potion required for a Good array.

Constraints:

$1 \leq N \leq 10^5$

$1 \leq K \leq 10^{12}$

$0 \leq A_i \leq 10^9$

Note:

A Good Array will always exist.

SAMPLE INPUT

```
3 3
1 2 3
```

SAMPLE OUTPUT

```
2 1
```

Explanation

The two possible Good arrays are: {1, 2} and {3}. Magic Potion required to form {1, 2} is 1 while required to form {3} is 2.

So the final answer is 2 and 1.

Time Limit:	1.0 sec(s) for each input file.
Memory Limit:	256 MB
Source Limit:	1024 KB
Marking Scheme:	Marks are awarded when all the testcases pass.
Allowed Languages:	C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Visual Basic

CODE EDITOR

Enter your code or Upload your code as file. Save C++ (g++ 5.4.0)

```
204             ans2 = min(ans2,(n-1-maxIndex) + (i+1));
205
206         }
207         if(ans2==n)
208         ans2=0;
209         cout<<ans<<" "<<ans2<<endl;
210         return 0;
211     }
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
```

```
232
233
234
235
236
237
238
239
240
241
242
```

243:1

Provide custom input

Press Ctrl-space for autocomplete suggestions.

COMPILE & TEST SUBMIT

Submission ID: 16693358 / 5 seconds ago

RESULT: Accepted

Score	Time (sec)	Memory (KiB)	Language
30.0	0.77143	64	C++

Input	Result	Time (sec)	Memory (KiB)	Score	Your Output	Correct Output	Diff
Input #1	✓	0.10965	64	10			
Input #2	✓	0.109119	64	10			
Input #3	✓	0.111139	64	10			
Input #4	✓	0.109739	64	10			
Input #5	✓	0.110051	64	20			
Input #6	✓	0.111094	64	20			
Input #7	✓	0.110635	64	20			

Compilation Log

No compilation log for this submission.

?

Your Rating:

COMMENTS (11)

Join Discussion...

Cancel Post

View all comments

PROGRAMMERS WHO SOLVED THIS PROBLEM ALSO SOLVED

The String Monster

Attempted By: **543** / Accuracy: **75**

53 Votes

Choose K Numbers

Attempted By: **1069** / Accuracy: **23**

3 Votes

Timely Orders

Attempted By: **1784** / Accuracy: **19**

520 Votes

?