Weird Faculty

This semester you are taking a course taught by a faculty member who has a weird way of grading tests. In a test, *n* questions will be asked, and the correctness of the answers is already determined. For the i^{th} question, the verdict will be *v[i] (where 0 ≤ i < n)*. If *v[i] = 1*, the answer is correct but if *v[i] = 0*, the answer is wrong.

When a test is given, you have to answer the first *k* questions *(indices 0 to k-1)* where *0 ≤ k < n*, and your friend has to answer the remaining questions *(indices k to n-1)* on the test. At the end, results are calculated as follows:

```
Your results:

int Your_result = 0;
for(int i=0;i<k;i++)
{
    if(v[i]==1)
        Your_result++;
    else Your_result--;
}
```

```
Your friend's results:

int YourFriend_result = 0;
for(int i=k;i<n;i++)
{
    if(v[i]==1)
        YourFriend_result++;
    else YourFriend_result--;
}
```

Choose the minimum *k* such that *Your_result > YourFriend_result*.

Function Description

Complete the function *exam* in the editor below. The function must return an integer that denotes the minimum number of questions you must answer to have *Your_result > YourFriend_result*.

exam has the following parameter(s):

v[v[0],...v[n-1]]: an array of integers

Constraints

- $1 \le n \le 10^5$
- $v[i] \in \{0,1\}$ (where $0 \le i < n$)

Input Format For Custom Testing

Sample Case 0

Sample Input 0

```
5
1
0
0
1
0
```

Sample Output 0

```
0
```

Explanation 0

n = 5

v = {1, 0, 0, 1, 0}

If you answer *0* questions *(k=0)* then *Your_result = 0* and *YourFriend_result = -1* (*2* correct answers & *3* wrong answers).

Sample Case 1

Start tour

For help on how to read input and write output in Python 3, click here.

Original Code | Python 3

```
17
18  def value(e):
19      if e == 1:
20          return 1
21      else:
22          return -1
23
24  def exam(v):
25      me = 0
26      my_friend = 0
27
28      for e in v:
29          my_friend += value(e)
30
31      k = 0
32      while k < len(v)-1:
33          if (me > my_friend):
34              return k
35
36          me += value(v[k])
37          my_friend -= value(v[k])
38          k += 1
39
40      return k
41
42
43  if __name__ == '__main__':…
```

Line: 10 Col: 1

Test against custom input

Run Code | Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

Compiled successfully. All available test cases passed!

Tip: Debug your code against custom input

Test Case #1: ✔	Test Case #4: ✔	Test Case #7: ✔
Test Case #2: ✔	Test Case #5: ✔	Test Case #8: ✔
Test Case #3: ✔	Test Case #6: ✔	

Testcase 1: ***Success***

Input [Download]

```
5
1
0
0
1
0
```

Expected Output [Download]

```
0
```

Testcase 2: *Success*

Input [Download]

```
5
1
1
1
0
1
```

Your Output

```
2
```

Expected Output [Download]

```
2
```

Testcase 3: *Success*

Input [Download]

```
100
0
1
1
0
0
0
0
0
0
1
1
1
0
1
0
0
0
0
0
1
0
1
0
0
1
0
1
0
1
1
0
1
0
0
1
1
0
0
0
0
1
0
1
1
1
```

1
2
3
4
5
6

```
1
0
1
1
1
1
0
0
0
1
0
0
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
```

Your Output

```
84
```

Expected Output [Download]

```
84
```

Testcase 4: *Success*

Your Output

```
Output hidden
```

Testcase 5: *Success*

Your Output

```
Output hidden
```

Output hidden

Testcase 7: ***Success***

Your Output

```
Output hidden
```

Testcase 8: ***Success***

Your Output

```
Output hidden
```