Color rocks
1 second, 256 MB

You are given a sequence of ***N*** color rocks. The **i**-th rock has color C_i where C_i is an integer. After you receive each rock, you put them at the top of a vertical pile. It turns out that if there are ***K*** consecutive rocks of the same color in the pile, they will magically disappear. (**Note:** In this task, ***K*** can be 2 or 3 depending on test cases. The case where ***K*** = **2** is simpler, and you can score 50% of the points in this task by working only on this case.)

Consider the following example. Suppose that ***N*=11** and ***K*=2**. Suppose that the rocks that you get have the following colors (in order): 1, 2, 4, 4, 3, 5, 5, 2, 2, 3, 9. This is what is going to happen in your pile of rocks.

Steps:	1	2	3	4	5	6	7	8	9	10	11
							5		2		
				4		5	5	2	2	3	
			4	4	3	3	3	3	3	3	9
		2	2	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1	1

Note that in steps 4, 7, 9, and 10, K consecutive rocks of the same colors disappear (shown in orange). In the end you have 3 rocks left.

Consider another example where K=3 and N=11. Suppose the the sequence of rocks are of the following colors: 1, 1, 2, 5, 5, 4, 4, 4, 5, 2, 2.

Steps:	1	2	3	4	5	6	7	8	9	10	11
								4			
							4	4			
						4	4	4	5		
					5	5	5	5	5		2
				5	5	5	5	5	5	2	2
			2	2	2	2	2	2	2	2	2
		1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1

In this example, you have 2 rocks left.

You task is to write a program that takes the sequence of color rocks and find out **the number of rocks left** after the process. In examples above, the answers are 3 and 2, consecutively.

Input
The first line contains two integers ***N*** and ***K*** (1<=***N***<=100,000; 2<=***K***<=3). **For 50% of test cases, *K*=2.**

The second line contains ***N*** integers representing the colors of the rocks. More specifically, for 1<=***i***<=***N***, the ***i***-th integer C_i is the color of the ***i***-th rock (1<=C_i<=1,000,000).

Output
You program should output one integer: the number of rocks in the pile at the end of the process.

Example 1

Input	Output
11 2 1 2 4 4 3 5 5 2 2 3 9	3

Example 2

Input	Output
11 3 1 1 2 5 5 4 4 4 5 2 2	2