

Guard wall sequences

Input:

outposts $\in \mathbb{N}$ , guards $\in \mathbb{N}$ , num[1...guards]  $\in \mathbb{N}^{guards}$

output:

count $\in \mathbb{N}$

precondition:

$1 < \text{outposts} < 100$ ,  $1 < \text{guards} < \text{outposts}$ ,  $\forall [i](1 < i < \text{guards}): 1 < \text{num}[i] < \text{outposts}$

Postcondition:

Count:= $\sum_{i=1}^{guards} 1$

( num[i+1]-num[i]<=2)

Algorithm		
cnt:=0		
i:=1...length (x)		
A(x[i])		
T		
F		
cnt:=cnt+1	—	

pattern		task
length (x)	$\rightarrow$	guards
cnt	$\rightarrow$	count
x[i]	$\rightarrow$	num[i+1]-num[i]<=2

guard wall sequences	
In:outposts,guards,num[]	
max:=num[0]	
count:=0	
n:=0	
index:=0	
i:=1...guards	
num[i]>num[i+1]	
T	F
n=num[j]	
num[j]=num[j+1]	
num[j+1]=n	
num[index+1]-num[index]<=2	
T	F
count ++	
index++	
Out:=count	

Code:

```
using System;
namespace 笑死 115
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string input = Console.ReadLine();
            int outposts = Convert.ToInt32((input.Split(" ")[0]));
            int guards = Convert.ToInt32((input.Split(" ")[1]));
            int[] num = new int[guards];
            int count = 0;
            for (int i = 0; i < guards; i++)
            {
                num[i] = Convert.ToInt32((Console.ReadLine()));
                int max = num[0];
            }
            for (int i = 0; i < num.Length - 1; i++)
            {
                for (int j = 0; j < num.Length - 1; j++)
                {
                    if (num[j] > num[j + 1])
                    {
                        int n = num[j];
                        num[j] = num[j + 1];
                        num[j + 1] = n;
                    }
                }
            }
            int Index = 0;

            while(Index<num.Length - 1)
            {
                while(Index < num.Length - 1 && num[Index + 1] - num[Index] <=2 )
                {
                    Index++;
                }
                count++;
                Index++;
            }
            Console.WriteLine(count);
            Console.WriteLine();
        }
    }
}
```