Domains Contests Rank Leaderboard Jobs nachomonllor

All Domains > Algorithms > Implementation > Lisa's Workbook

Badge Progress (Details)

Points: 864.09 Rank: 25758

Lisa's Workbook

by Errichto

Problem Submissions Leaderboard Discussions Editorial

Lisa just got a new math workbook. A workbook contains exercise problems, grouped into chapters.

- There are n chapters in Lisa's workbook, numbered from 1 to n.
- The i-th chapter has t_i problems, numbered from 1 to t_i.
- Each page can hold *up to* k problems. There are no empty pages or unnecessary spaces, so only the last page of a chapter may contain fewer than k problems.
- Each new chapter starts on a new page, so a page *will never* contain problems from more than one chapter.
- The page number indexing starts at 1.

Lisa believes a problem to be *special* if its index (within a chapter) is the same as the page number where it's located. Given the details for Lisa's workbook, can you count its number of *special* problems?

Note: See the diagram in the *Explanation* section for more details.

Input Format

The first line contains two integers n and k — the number of chapters and the maximum number of problems per page respectively.
The second line contains n integers $t_1, t_2, \ldots, t_n$, where t_i denotes the number of problems in the i-th chapter.

Constraints

- $1 \leq n, k, t_i \leq 100$

Output Format

Print the number of *special* problems in Lisa's workbook.

Sample Input

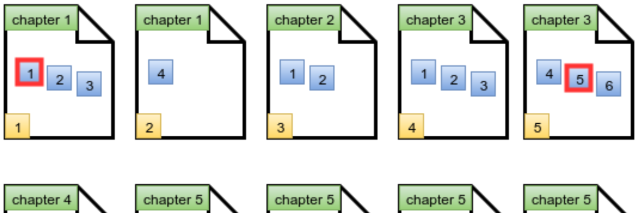
```
5 3
4 2 6 1 10
```

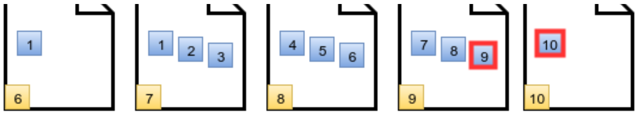
Sample Output

```
4
```

Explanation

The diagram below depicts Lisa's workbook with $n = 5$ chapters and a maximum of $k = 3$ problems per page. Special problems are outlined in red, and page numbers are in yellow squares.





There are 4 special problems and thus we print the number 4 on a new line.

Submissions: 16482
Max Score: 25
Difficulty: Easy

Rate This Challenge:
☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable)

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

class Solution {
    static void Main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be
named Solution */

         string[] input = Console.ReadLine().Split(' ');
            int n = int.Parse(input[0]);
            int k = int.Parse(input[1]);

            int[] ppcap = Array.ConvertAll(Console.ReadLine().Split(' '), e => int.Parse(e));

            //int n = 5, k = 3;
            //int[] ppcap = { 4, 2, 6, 1, 10 };

            List<string> paginas = new List<string>();

            for (int i = 0; i < ppcap.Length; i++)
            {
                int prob_cap = ppcap[i];

                string concat = "";
                for (int j = 1; j <= prob_cap; j++)
                {

                    if (j%k == 0)
                    {
                        concat += j;
                        paginas.Add(concat);
                        concat = "";
                    }
                    else
                    {
                        concat += j + ",";
                    }
                }
                if (concat.Length > 0)
                {

                    paginas.Add(concat);
                }

            }

            //foreach (string s in paginas)
            //{
            //    Console.WriteLine(s);
            //}
            int ans = 0;
            for (int indicePagina = 0; indicePagina < paginas.Count; indicePagina++)
            {
                string[] cap = paginas[indicePagina].Split(',');
                if (cap.Contains((indicePagina + 1).ToString()))
                {
                    ans++;
                }
            }
            }

            Console.WriteLine(ans);

    }
}
```

Line: 4 Col: 19

Upload Code as File □ Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

✓ Test Case #0
✓ Test Case #3
✓ Test Case #6
✓ Test Case #9
✓ Test Case #1
✓ Test Case #4
✓ Test Case #7
✓ Test Case #10
✓ Test Case #2
✓ Test Case #5
✓ Test Case #8

Next Challenge

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature