Problem: Lisa’s Workbook

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

- There are chapters in Lisa's workbook, numbered from to .
- The -th chapter has problems, numbered from to .
- Each page can hold *up to* problems. There are no empty pages or unnecessary spaces, so only the last page of a chapter may contain fewer than 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 .

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 and — the number of chapters and the maximum number of problems per page respectively.

The second line contains integers , where denotes the number of problems in the -th chapter.

Constraints

-

Output Format

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

Sample Input

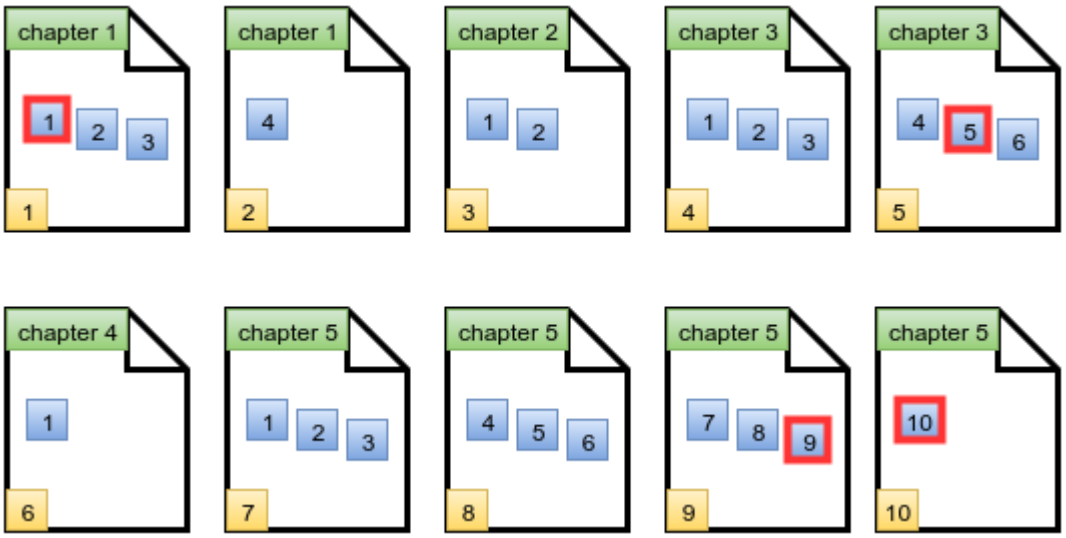
```
5 3
4 2 6 1 10
```

Sample Output

```
4
```

Explanation

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



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

Solution

```
int chapters, maxP, page=1, counter=0;
   cin>>chapters>>maxP;  //maxP --> maximum questions per page
   int array[chapters];

   /*Necessary data feeding loop*/
   for(int i=1; i<=chapters; i++)  //runs for each chapter
           {                                  //separately and inputs data
               cin>>array[i];        //Inputs the data
               int inc=0;        //inc--> how many pages will be used ?
               (array[i]%maxP!=0  ?  inc=array[i]/maxP+1  :  inc=array[i]/maxP);
                                                  //value for inc
                int que=1,
                for(int j=page; j<page+inc; j++)    //runs separately for
                                  each page of this chapter
                      {    //Runs per question on each page
                           while(que<=maxP*(j-page+1)  &&  que<=array[i])
                               {
                                       if(que==j)  //page is special ?
                                                {counter+=1;}
                                       que++;    //moves on to next page for check
                                }
                      }
             page=page+inc;  //updates page value
           }
       cout<<counter;  //prints the answer
    }
}
```

- Anshul AgGarwal