# Brunhilda's Birthday

Except for her affinity towards old armours, Brunhilda is a normal seven year old girl. Thus, she is planning the perfect birthday party, for which she has invented the following game: All children run around until some number $k$ is announced. Then all children try to form groups of exactly $k$ people. As long as at least $k$ children are left over, further groups of $k$ children are formed. In the end, less than $k$ children are left over and will be eliminated (not literally of course) from the game. The game continues with further numbers announced, and ends if all children are out.

Brunhilda asked her father Wotan to announce the numbers in the game. Wotan does not like this game and announced $\infty$ when they first tried it. Brunhilda thinks this would be quite embarrassing at the party, and so she gave him a list of $m$ prime numbers from which he can choose for each call; he may use the same number more than once.

Wotan would like to end the game as soon as possible since he has tickets for a match of his favourite football club FC Asgard. Unfortunately, Brunhilda does not know the number of children at her party yet. Now, for $Q$ different numbers $n_1, \ldots, n_Q$ of children, Wotan wants to know in advance the least number of calls he will need to end the game.

## Input

The first line contains the integers $m$ and $Q$ described above. The second line contains $m$ different prime numbers $p_i$ ($1 \leq i \leq m$) in ascending order: the list of prime numbers Wotan can use. The following $Q$ lines contain one integer $n_j$ ($1 \leq j \leq Q$) each: the number of children who might take part in the game.

## Output

The output should consist of $Q$ lines. The $j$-th line should contain the answer for $n_j$: if Wotan can end the game it should contain the least number of calls he needs (an integer), otherwise the line should contain the string oo (two lower case letters o, meaning $\infty$).

## Constraints

$1 \leq m \leq 100\,000, 1 \leq Q \leq 100\,000, 2 \leq p_i \leq 10\,000\,000, 1 \leq n_j \leq 10\,000\,000$

In test cases worth 20 points: $m, n_j, Q \leq 10\,000$

In test cases worth additional 20 points: $Q = 1$

## Example

| Input | Output |
|---|---|
| 2 2<br>2 3<br>5<br>6 | 3<br>oo |

## Limits

| | |
|---|---|
| **Time limit:** | 1 sec per test case |
| **Memory limit:** | 256 MB per test case |

# Tracks in the Snow

There is a rectangular meadow in a forest, having been covered with a blanket of fresh snow in the morning (left in the figure below).

Rabbits and foxes, who live in the forest, are crossing the meadow and leave their tracks in the snow. They always enter in the upper left corner and leave the meadow from the lower right corner. In between they can move back and forth, playing in the snow, even crossing their own tracks. At any time there is at most one animal on the meadow. No animal enters the meadow more than once. The movements of the animals can be described by dividing the meadow into quadratic cells. The animals never move diagonally in a single step and they never jump over a cell. When an animal enters a cell its tracks will cover all previous tracks left in this cell.

For example, first a rabbit crossed the meadow from top-left to bottom-right (middle in the figure). After that, a fox crossed, and now his tracks are partially covering the rabbit's (right in the figure).

```
........   RRR.....   FFR.....
........   ..RRR...   .FRRR...
........   ..R.....   .FFFFF..
........   ..RRRR.R   ..RRRFFR
........   .....RRR   .....FFF
```

You are given a map of the meadow at some time after indicating for each cell if there are any visible tracks and whether they were left by a rabbit or by a fox (right in the figure). You are interested in the local wildlife population. Write a program to determine the minimal possible number $N$ of animals that must have crossed the meadow to leave the given pattern of tracks in the snow.

## Input

The first line contains two integers $H$ and $W$, the height and the width of the map of the meadow. $H$ lines follow with exactly $W$ characters on each: the map, where '.' marks untouched snow, 'R' a spot where a rabbit's track is the topmost one, and 'F' a spot where a fox's track is the topmost one. There is at least one track on the meadow.

## Output

The output should consist of a single integer: the minimal number $N \geq 1$ of animals that could have left the tracks given in the input.

## Constraints

$1 \leq H, W \leq 4000$

In test cases worth 30 points: $N \leq 200$ and $H, W \leq 500$

## Example

| Input | Output |
|---|---|
| 5 8<br>FFR.....<br>.FRRR...<br>.FFFFF..<br>..RRRFFR<br>.....FFF | 2 |

## Limits

**Time limit:** 2 sec per test case
**Memory limit:** 1300 MB per test case

# Vim

Ernest Vincent Wright was an American author, known for writing a novel (Gadsby) without using the letter "e". Victor is a big fan of Ernest and tries to imitate him in writing a novel, but is looking for a real challenge. He uses only the first ten characters of the alphabet (namely abcdefghij). Ironically, the "e" key on his computer breaks halfway through the novel, and for consistency, he decides to delete all the "e"s he has already written. His friend, a programmer, recommended him to use the text editor Vim to perform this task. Unfortunately, Victor is not very familiar with Vim, and knows only three different commands: "x", "h" and "f".

- "x" deletes the character at the cursor. The cursor position (counted from the left) does not change. Victor shall not use this command if the cursor is at the last character of the document.
- "h" moves the cursor one step backward (to the left). Nothing happens if the cursor is at the beginning of the document.
- "f" waits for the user to input another character $C$, and then moves the cursor forward to the next occurrence of $C$ (even if the character at the cursor happens to be $C$). Nothing happens if $C$ does not occur anywhere to the right of the cursor position.

For example, if the current text is

$$\texttt{jeff}\boxed{\texttt{i}}\texttt{ehadabigidea}$$

where the cursor is denoted by a frame $\boxed{\phantom{x}}$, then

- "x" would give $\texttt{jeff}\boxed{\texttt{e}}\texttt{hadabigidea}$
- "h" would give $\texttt{jef}\boxed{\texttt{f}}\texttt{iehadabigidea}$
- "fi" would give $\texttt{jeffiehadab}\boxed{\texttt{i}}\texttt{gidea}$

Write a program that calculates the least number of key presses that Victor needs to use to delete all the "e"s in the document, but no other letters. Initially, the cursor is at the first character of the document. Note that the "e" key is broken, so the command "fe" cannot be used.

## Input

The first line contains the integer $N$, the length of the document. The next line contains $N$ characters, each one of the ten lowercase letters from "a" to "j". The first and the last letter of the input are both different from "e".

## Output

The only line of output should contain exactly one integer: the least number of key presses Victor needs to delete all the "e"s.

## Constraints

$N \leq 70\,000$

In test cases worth 50 points: $N \leq 500$

In test cases worth additional 10 points: $N \leq 5\,000$

## Example

| Input | Output |
|---|---|
| `35`<br>`chefeddiefedjeffeachbigagedegghehad` | `36` |

An optimal solution for the example test case is:

```
fdhxhhxffhxfahxhhhxhhhxfdhxfghxfahhx
```

You can test this by starting the Vim editor yourself (type “`vim file.txt`” at the command prompt to open `file.txt`, type “`:q<ENTER>`” to quit).

## Limits

**Time limit:** 2 sec per test case
**Memory limit:** 512 MB per test case