French-Australian Regional Informatics Olympiad

11th March, 2011

Duration: 4 hours

4 questions

All questions should be attempted

Problem 1
Bookshelf

Input File: *standard input*
Output File: *standard output*

Time and Memory Limits: 1 second, 32 MB

In the town of Harigato, there is a library that holds a collection of books on the ancient order of the Harigatan monks. These books are stored side-by-side in a long line, and not all books are of the same height. Some of these books are irreplaceable as they are thousands of years old.

Unfortunately, an outbreak of mildew has begun in the collection, destroying the books that are infected with mildew. Mildew infections begin with one or more infected books. If a shorter book is next to an infected book, it will be safe from the mold. However, books at the same height or taller than an adjacent infected book will also become infected.

The library would like to find out how such outbreaks would affect their collection and they have turned to you for help. Given the order and heights of books along the shelf, which books are irreplaceable, and which books start off with a mildew infection, your task is to find out how many irreplaceable books would be lost.

Input

The first line of the input will contain three positive integers, N M I, representing the total number of books on the shelf, the number of books affected by mildew and the number of books that are irreplaceable, respectively. There will be at most 100 000 books on the shelf.

The second line of the input will contain N positive integers, specifying the height of the N books. The heights are all less than 1 000 000. The books sit on the shelf in the order given and are numbered from $1 \ldots N$.

The third line of the input will contain M integers, specifying which books have been infected with mildew. Each of these integers will be between 1 and N. No book will appear more than once in this list.

The fourth and final line of input will contain I integers, specifying which books are irreplaceable. Each of these integers will be between 1 and N. No book will appear more than once in this list.

Output

Your output should consist of a single integer – the number of irreplaceable books which would be infected with mildew in the given scenario.

Sample Input

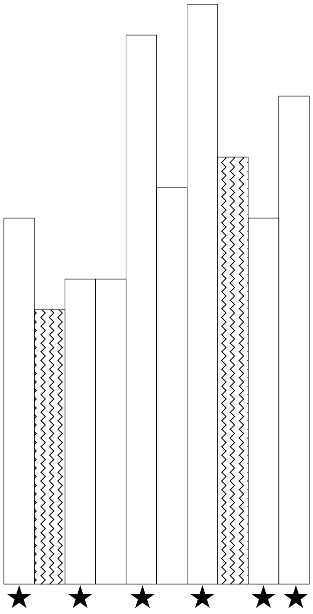
```
10 2 6
12 9 10 10 18 13 19 14 12 16
2 8
1 3 5 7 9 10
```

Sample Output

```
4
```

Explanation

The sample data corresponds to the bookshelf below. The books infected with mildew are shaded. The irreplaceable books are marked with a star. The mildew from book 2 can spread to irreplaceable books 1, 3 and 5. The mildew from book 8 can spread only to irreplaceable book 7. Thus there are a total of 4 irreplaceable books infected with mildew after the infection has spread.



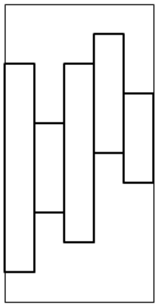
Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.

Problem 2
House of a Thousand Blades

Input File: *standard input*
Output File: *standard output*

Time and Memory Limits: 1 second, 32 MB



Seiko is a student of the *katana* (sword). She practises from morning to night under the watchful eye of her teacher, an old man who offers her deep wisdom hidden in cryptic sayings.

Today her teacher takes her to the *koi* pond and produces a flat wooden board. He carefully paints a design on the board: a series of vertical stripes of equal width. The right edge of each stripe touches the left edge of the next, with the whole design forming one contiguous shape that spans the width of the board. One such masterpiece is shown to the right.

The teacher then explains to Seiko today's lesson:

> "Elegance. The swordsmaster wastes neither time nor effort. Each movement of your blade must be definite and imbued with purpose. Take your sword and carve out the shape I have painted on this board, so that it becomes a single detached piece."

Seiko's swordwork is precise, and in one cut she can carve through any straight line segment of the board, regardless of the angle of the line segment or whether its endpoints have already been cut.

Meditating on her teacher's words, Seiko realises that her task now is to find the smallest number of cuts she needs to carve out the painted shape.

Input

The first line consists of two space separated integers, W and H ($1 \leq W, H \leq 100\,000$), representing the width and height of the wooden board respectively. The board can be thought of as a rectangle with co-ordinates ranging from (0,0) to (W, H).

The second line consists of W integers describing the top edges of the W brush-strokes. Each height will be between 1 and H inclusive.

The third line consists of W integers describing the bottom edges of the W brush-strokes. Each height will be between 0 and $H - 1$ inclusive. It is guaranteed the top edge of every brush-stroke is strictly higher than its bottom edge, and that all the brush-strokes form one connected shape, i.e. for all $i > 1$, $bottom_i < top_{i-1}$ and $bottom_{i-1} < top_i$.

For 40% of the available marks, $N \leq 1\,000$.

Output

You should output one line with one integer – the minimum number of cuts needed to obtain the desired shape from the board.

Sample Input

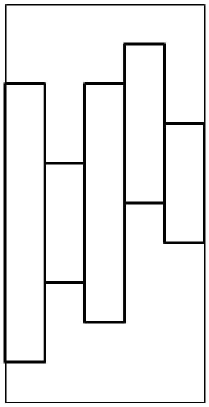
```
5 10
8 6 8 9 7
1 3 2 5 4
```

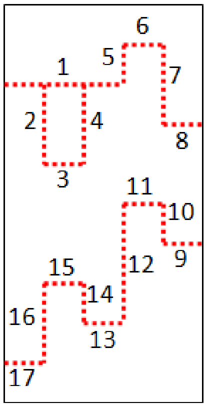
Sample Output

```
17
```

Explanation

The sample data corresponds to the board shown below on the left. The 17 cuts are shown on the right.





Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.

Problem 3
Life

Input File: *life*.k.*in*
Output File: *life*.k.*out*

Output Only Task

Today is the most important day of your career as an exobiologist. After many attempts, you finally managed to maintain a stable micro-wormhole for more than 20 nanoseconds between your lab and PHR 5426b, the most earth-like exoplanet discovered to this day, and were even able to repeat the experiment 10 times before the plasma of your hypergenerator collapsed. 20 nanoseconds is just enough time for enough photons to traverse the wormhole and be captured by the camera of your xPhone 5.

Using a $49 xPhone 5 to capture the most important images of the century may seem odd, but the xxd sensor in its camera is by far the most sensitive ever manufactured, and is much better than even the $500,000 sensors created for Hubble 5.

However, just after announcing the big discovery to the world and organising the largest press conference since the launch of the xPhone 5 itself, you realize that you have a problem: since the wormhole is really tiny, you had to take many pictures to get a whole image of each life form. Unfortunately, you were so excited about your discovery that you forgot to keep track of which picture corresponds to which part of the whole image. You absolutely need to rearrange them before the press conference starts.

Input

You are given ten input files `life.`k`.in` ($1 \leq k \leq 10$), describing the ten scrambled pictures.

The first line of each file contains four integers, W H C R, indicating that the image is composed of a grid of $C \times R$ tiles (C columns and R rows), each tile being a small image of $W \times H$ pixels (W columns and H rows).

Following this are $C \times R$ groups of H lines. Each group describes a tile with H lines of W integers between 0 and 255 inclusive, giving the brightness of the pixels that compose the tile. The tiles are numbered $1 \ldots (C \times R)$ and are given in this order. Your only problem is that this order is completely unrelated to the position of each tile in the original image.

Your task is to determine in which order the $C \times R$ tiles need to be arranged to form an image as similar as possible to the original image. Since you don't have the original images, you can only rely on the fact that these are images of life forms on a distant planet.

Output

For each input file `life.`k`.in`, you must produce an output file `life.`k`.out` that contains your solution.

Each solution file should contain R lines of C distinct integers, each being the number of a tile. These numbers should be arranged in such a way that placing the corresponding tiles in this manner on an $C \times R$ grid creates an image that looks as much as possible like an alien form of life.

Sample Input

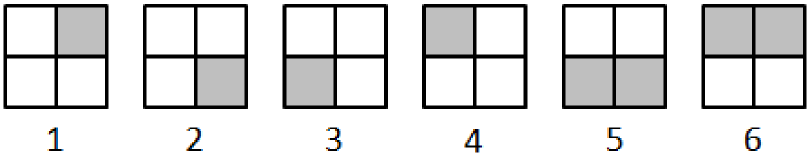
```
2 2 3 2
0 128
0 0
0 0
0 128
0 0
128 0
128 0
0 0
0 0
128 128
128 128
0 0
```

Sample Output

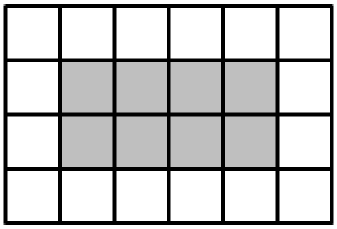
```

5 1 2
6 4 3
```

Input Details

The input describes 6 tiles, as illustrated below:



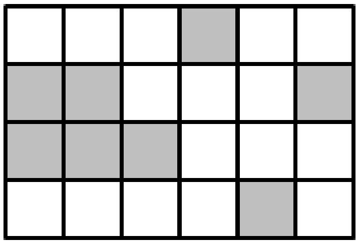
The original image of this very simple life form is illustrated below:



The above sample case is not representative of the test cases provided.

Output Details

The output describes an ordering of the tiles that corresponds to the following recomposed image:



Scoring

Your score for each input scenario will be determined as follows:

- If your solution doesn't contain exactly R lines of C integers, all distinct and between 1 and $R \times C$, it will obtain a score of 0%.
- Otherwise, let A be the total number of adjacent pairs (sharing one side) of tiles in the image ($A = C \times R \times 2 - C - R$), and let B be the number of adjacent pairs of tiles in your output that are adjacent in the same way (same sides) as in the original image. Your solution will be scored $100 \times B/A$ and rounded down (expressing a percentage of points).

For example, in the sample case above, a perfect output file would be:

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

As this corresponds exactly to the original image, all 7 pairs of adjacent tiles are correct, so this output would have given a score of 100% ($100 \times 7/7$).

In the sample output, however, only the pairs $(5, 6)$ and $(6, 4)$ are adjacent in the same way as in the original image. This output would thus be awarded a score of 28% ($100 \times 2/7 = 28$).

Experimentation

Use the trial runs feature on the submission site in order to experiment with the input data. There you may submit a possible mapping, and view the resulting image.

Submitting for an Output Only Task

To submit your output files, you must do the following:

- Create a zip file containing all of your output files. You may include as many or as few output files as you like (i.e., you do not need a solution for every input scenario). On a GNU/Linux system you can use a command like the following:

  ```
  zip mysolutions.zip life.*.out
  ```

  On Windows systems you can create a zip file by selecting *File* → *New* → *Compressed (zipped) Folder* from within Windows Explorer, and then you can copy your output files into this new zip file.
- Submit this zip file to the FARIO website, in the same way that you would submit an ordinary solution.

The website will look inside the zip file and ensure that the output files are named correctly, although it will not check the formatting or layout of these files.

If you resubmit a different zip file, **the old zip file will be deleted**. For instance, suppose you submit a zip file with solutions to the first nine scenarios, and later on you find a solution to the tenth. Your new submission must contain **all ten output files**, since when you submit your new solutions the old ones will be thrown away.

Please contact the contact organisers at `fario@orac.amt.edu.au` if you have any questions about this procedure.

Problem 4
The Virus

Input File: *standard input*
Output File: *standard output*

Time and Memory Limits: 1 second, 32 MB

No one knows how it all started, just that it happened so quickly. Two weeks ago you were watching the news about political problems in far away nations, today you are on the run from your own neighbours who have fallen victim to *The Virus*. There is no known cure for *The Virus* which is plaguing the nation – once a person has contracted it, they cannot be cured. Thankfully though, Dr. Yes has come to you with a curious biological weapon, which is used to deploy a vaccine against the virus. Using his weapon, the vaccine can be administered to everybody within a city instantly, effectively stopping the spread of *The Virus* within that city.

The eccentric Dr. Yes hastily describes to you in a raspy voice his plan to save the nation. Together you will travel to every city in the country and use the weapon on each city, saving those you can by vaccinating the healthy individuals against *The Virus*.

Each city's mayor has determined the rate of spread of *The Virus* in their city, measured in victims per hour. You need to perform this rescue option to minimise the number of victims. Some roads take longer to travel on than others, and each wasted hour means more people lost to *The Virus*. Given a map of all the roads in the country and the number of citizens in each city falling victim to *The Virus* every hour, determine a route passing through every city such that the total number of people infected by *The Virus* is as small as possible. Furthermore, in these dangerous times with thieves and murderers abound, you must not travel on any road more than twice.

There exist N cities in the country, and $N-1$ bi-directional roads which connect the cities such that there is a path from every city to every other city. You know from your days as an informatics student that the graph formed is called a "tree". You also may assume that city populations are sufficiently large that they will never lose all their citizens to *The Virus*.

You are about to start coding, when Dr. Yes interrupts you with a final remark.

> "Remember, lives are in your hands. Code fast and we will be the heroes of tomorrow."

Input

The first line of input will contain a single integer N, the number of cities ($1 \leq N \leq 100\ 000$). The cities are numbered from 1 to N. You must commence your rescue operation in city 1.

The second line of input will contain N non-negative integers. The ith integer represents the number of citizens succumbing to *The Virus* each hour in city i. This value will be at most 100.

This will be followed by $N-1$ lines of input. Each line will contain three space-separated integers a, b and h, representing a road from city a to city b that takes h hours to travel ($1 \leq h \leq 100$).

For 60% of the available marks, $N \leq 1\ 000$.

For 20% of the available marks, $N \leq 10$.

Output

Output should consist of one line with a single integer: the minimum number of victims of *The Virus* before every city is saved.

Sample Input

```
5
9 10 2 5 1
1 2 2
1 4 4
3 4 3
4 5 5
```

Sample Output

```
101
```

Explanation

The optimal path is:

$$1 \xrightarrow{\text{2 hours}} 2 \xrightarrow{\text{2 hours}} 1 \xrightarrow{\text{4 hours}} 4 \xrightarrow{\text{3 hours}} 3 \xrightarrow{\text{3 hours}} 4 \xrightarrow{\text{5 hours}} 5$$

City	Hours until saved	Infected each hour	Citizens lost to *The Virus*
1	0	9	0
2	2	10	20
3	11	2	22
4	8	5	40
5	19	1	19
			Total: 101

Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.