
**French-Australian Regional
Informatics Olympiad
13th March, 2009**

Duration: 4 hours

4 questions

All questions should be attempted

Problem 1

Bookshop

Time and Memory Limits: 1 second, 16 Mb

You are the author of a once world-famous book series with a plot revolving around wizards, flying broomsticks and disgruntled dark lords. In their prime you sold millions of books. At one point, every self-respecting teenager had the complete series. The secret of your success, never revealed to anyone, is that each book you released had a strictly greater width, measured in number of pages, than your previous book. Sadly, as time has passed on, you have slowly faded to obscurity, your once-famous books all but forgotten.

On one particularly dull day you find yourself in an antique book store. The store has a single long shelf filled with countless books, all of different widths. The books on the shelf are kept sorted in chronological order, with the oldest books at the left-hand side of the shelf and the newest books at the right-hand side.

In a fit of nostalgia, you decide to find the copies of your once-famous books on the shelf. So much dust has accumulated on the books that it is impossible to make out the titles by trying to read their spines. Fortunately, all the books on the shelf have different widths, and you happen to remember the precise width of each of the books you have published, allowing you to hunt for them.

Your task is to find the location of each of your books on the bookshelf. You are given the list of the widths of each of the N books on the bookshelf, ordered from the oldest book to the newest book. Additionally, you are given a list of the widths of each of the K books you have published, also ordered from oldest to newest. No two books on the shelf have the same width, and you are certain that the shelf contains one copy of each of your books.

Constraints

- $1 \leq N \leq 100\,000$, where N is the number of books on the bookshelf.
- $1 \leq K \leq N \leq 100\,000$, where K is the number of books you have written.
- $1 \leq A_i \leq 1\,000\,000\,000$, where A_i is the width of the i th book on the bookshelf.

Input

Your program must read from standard input. The first line of input will contain a single integer N , the number of books on the bookshelf. The following N lines will each contain a single integer A_i , describing the width of one book on the bookshelf. These books will be listed in chronological order (from left to right). No two books will have the same width.

The following line will contain a single integer K , the number of books that you have published. The following K lines will each contain a single integer B_j , describing the width of the j th book that you wish to find. Your books will be listed in chronological order (so the widths B_1, \dots, B_K will appear in sorted order in the list A_1, \dots, A_N), and each book's width will be strictly greater than the previous (so $B_1 < B_2 < \dots < B_K$).

Output

Your program must write K lines to standard output. The j th line of the output should contain a single integer describing the position on the bookcase of the j th book you have written. The first book on the bookcase has position 1, while the last book on the bookcase has position N .

Sample Input

9
815
223
251
97
317
1002
636
766
312
5
223
251
317
636
766

Sample Output

2
3
5
7
8

Explanation

In this example, you are standing in front of a bookshelf containing 9 books. Furthermore, you have published 5 books that you wish to find on the bookshelf. The first book you wish to find has width 223 and appears as the second book on the bookshelf, which means that the first location that you output is 2. Similarly, your remaining books of widths 251, 317, 636, and 766 appear in locations 3, 5, 7 and 8 on the store bookshelf respectively.

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

Toy Factory

Time and Memory Limits: 1 second, 16 Mb

To pursue your artistic destiny, you have taken up a job as a painter in a toy factory in a small village in France. Your dream is to become a famous painter, remembered with the likes of Renoir and Monet. Unfortunately, at this toy factory, your job is much more menial. The factory produces the famous flip-flap toy. Flip-flaps come in many different colours, though each individual flip-flap is painted a single colour all over.

You are placed at the end of the factory line. On one side, you have the flip-flaps coming off the line that require painting. Each flip-flap must be painted its designated colour. On the other side, you have C tins of paint—one for each possible colour. Your dreary job is to simply paint each toy the correct colour.

Painting the toy is easy if the tin of paint required is already open. If the required tin of paint is not already open, you must fetch your screwdriver and open the lid before you can paint the toy. In addition, company policy dictates that no more than K tins of paint may be open simultaneously, as excessive paint fumes lead to strange dreams. Once you have exactly K tins of paint open, you must close one tin before opening another. No tins of paint are open to begin with.

Because opening tins of paint is a messy and tedious task, you wish to avoid it as much as possible. Given the colours of the toys to paint and the order in which you must paint them, your task is to find the fewest number of times you need to open a tin of paint, in order to paint them all the correct colour.

Constraints

For 100% of the available marks:

- $1 \leq N \leq 100\,000$, where N is the number of toys you are required to paint.
- $1 \leq C \leq 100\,000$, where C is the number of tins of paint, each containing a different colour of paint.
- $1 \leq K \leq C$, where K is the maximum number of tins you may have open simultaneously.

Additionally, for 50% of these marks:

- $1 \leq N \leq 1000$.

Input

Your program must read from standard input. The first line of input will contain the integers $N C K$, as described above. Each colour is assigned an integer $1, 2, \dots, C$.

The following N lines will describe the colour of each toy on the factory line, in the order you are required to paint them. Each of these lines will contain a single integer c_i , the colour the i th toy must be painted ($1 \leq c_i \leq C$).

Output

Your program must write a single line to standard output. This line must contain a single integer, giving the fewest number of times you need to open a tin of paint.

Sample Input

10 4 2
 3
 4
 2
 2
 3
 4
 1
 4
 3
 4

Sample Output

6

Explanation

In this example, you have four tins of coloured paint but can only have up to two tins open at a time. In order to paint the 10 toys, one possible sequence of opening and closing tins is shown below.

Toy colour	Tins open	Action performed
3	3	Opened tin 3.
4	3, 4	Opened tin 4.
2	2, 3	Closed tin 4, opened tin 2.
2	2, 3	
3	2, 3	
4	3, 4	Closed tin 2, opened tin 4.
1	1, 4	Closed tin 3, opened tin 1.
4	1, 4	
3	3, 4	Closed tin 1, opened tin 3.
4	3, 4	

This sequence requires opening tins of paint six times. This is also the fewest number possible.

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

Memories

Time and Memory Limits: 2 seconds, 16 Mb

It is the start of a new year at Petit's School for Proper and Well-Balanced Children, and as the Chief Record Keeper it is your job to take the school photograph. Your training in the art houses of Europe has taught you that form is more important than content—with this in mind, you decide that it does not matter who is in the photograph as long as it looks good.

You decide to form R rows of children, with C children in each row. For each row, the *imbalance* of the row is the difference between the height of the shortest student in that row and the height of the tallest student in that row. The imbalance of the entire photograph is the largest imbalance seen in any of the rows.

There are N students in the school, all of whom are eager to have their photo taken. To save your reputation as both an artist and a well-balanced teacher, your task is to choose $R \times C$ of these students for your photograph so that the imbalance of the photograph can be made as small as possible.

As an example, suppose there are $N = 8$ students in the school, with the following heights (all measured in centimetres):

170, 205, 225, 190, 260, 130, 225, 160.

Suppose you need to form $R = 2$ rows of $C = 3$ students each. You can do this as follows:

Row 1: Heights 225, 205, 225
Row 2: Heights 160, 190, 170

The imbalance of row 1 is $225 - 205 = 20$, and the imbalance of row 2 is $190 - 160 = 30$. Therefore the imbalance of the entire photograph is $\max(20, 30) = 30$, which in fact is the smallest imbalance possible.

Constraints

For 100% of the available marks:

- $1 \leq R \leq 10\,000$ and $1 \leq C \leq 10\,000$, where R is the number of rows in your photograph and C is the number of students in each row.
- $R \times C \leq N \leq 1\,000\,000$, where N is the total number of students in the school.
- $1 \leq H \leq 1\,000\,000\,000$, where H is the height of a student as measured in centimetres. Something must be done about those high-energy school lunches.

Additionally, for 50% of these marks:

- $N \leq 5000$.

Input

Your program must read from standard input. The first line of input will contain the integers $N R C$, as described above. Following this will be N additional lines of input, each giving the height of a student (measured in centimetres).

Output

Your program must write a single line to standard output. This line must contain a single integer, giving the smallest possible imbalance for your photograph (again in centimetres).

Sample Input

```
8 2 3
170
205
225
190
260
130
225
160
```

Sample Output

```
30
```

Scoring

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

Problem 4

The World of Fragrance

Input File: *fragrance.k.in*

Output File: *fragrance.k.out*

Output Only Task

After some disappointing job experiences in the world of finance, you have decided to put your skills to a better use in the field of perfumery, where you now help to create the most exquisite perfumes. Creating new scents is an art that involves trying many different combinations of ingredients, and then using the perfumer's sense of smell to rate each new fragrance obtained.

Your new job is to help well trained perfumers save time, by generating only combinations of ingredients that are likely to smell good. You have collected data by asking the best noses in the world to rate various combinations of two ingredients at a time. Your theory is that if every pair of ingredients in a perfume smells good, then the perfume is likely to smell good overall.

You are provided with a total of N ingredients (numbered from 1 to N), and a list of P pairs of distinct ingredients with their corresponding ratings. A positive rating for a pair indicates a good smell, whereas a negative rating indicates a bad smell. Pairs of ingredients that have never been tested by experts are not listed and should be considered neutral, with a rating of 0.

The perfumers wish to create a new scent consisting of precisely K ingredients. Your task is to select a set of K distinct ingredients so that the sum of the ratings of every possible pair within this set is as high as possible. Note that you are not necessarily required to find the optimal solution. See the *Scoring* section for more details.

Constraints

- $2 \leq N \leq 1000$, where N is the number of ingredients available.
- $1 \leq K \leq 20$, where K is the number of distinct ingredients you need to include in your set.
- $1 \leq P \leq 100\,000$, where P is the number of pairs of ingredients that have been rated by experts.
- $-1000 \leq R_i \leq 1000$, where R_i is the rating of a given pair of ingredients.

Input

You are given ten input files, named `fragrance.M.in` ($1 \leq M \leq 10$).

The first line of each input file will contain the integers N K P , as described above. Each of the P remaining lines of the input file will contain three integers, A_i B_i R_i , indicating that the pair of ingredients (A_i, B_i) has been given the rating of R_i by the experts. It is guaranteed that $A_i \neq B_i$ for every such pair, and that no pair will be listed more than once.

Output

For each input file `fragrance.M.in`, you are required to produce a corresponding output file `fragrance.M.out` that describes your set of ingredients.

This output file should consist of precisely $K + 1$ lines. The first line must contain a single integer, the total of the ratings for all possible pairs of ingredients within your set. The next K lines must each contain a single integer describing one of the K distinct ingredients in your set.

Sample Input

```

5 3 7
1 2 12
1 3 10
1 5 -3
2 4 -2
2 5 -8
3 5 17
4 5 5

```

Sample Output

```

24
1
3
5

```

Explanation

In this example, you have five ingredients at your disposal, and must create a perfume using three of them. Seven pairs of these ingredients have been rated by experts, with ratings ranging -8 to 17 .

The best possible set for this example is the combination of ingredients 1, 3 and 5. All three possible pairs within this set have been rated by experts: $(1, 3)$ with a rating of 10 , $(1, 5)$ with a rating of -3 , and $(3, 5)$ with a rating of 17 , giving a total rating of $10 - 3 + 17 = 24$.

Scoring

There is no particular “best solution” that you are required to achieve. Instead, your score will be determined relative to the other contestants whom you are competing against (as well as the judges’ solution). It is guaranteed that there will be at least one solution with a positive overall rating for each input scenario.

For each input scenario, the contestant who achieves the largest overall rating will be identified. Suppose that this contestant achieves a perfume with overall rating r . Your score for this input scenario will then be:

- 100% if your program finds a solution also with overall rating r ;
- at least 10% if your program finds any valid solution;
- 0% if your program generates an incorrect solution (e.g. you incorrectly calculate the overall rating of your set);
- otherwise, determined by a formula relating your score against the best rating for the input scenario, as described below.

$$\text{Your score} = 10 + 90 \times \left(\frac{\text{your result}}{\text{best result}} \right)^5$$

For example, consider an input scenario for which the best solution finds a set with overall rating 40 . Then the scoring scale for a *correct* solution would be as follows:

Overall rating	0	5	10	15	20	25	30	35	40
Score	10%	10%	10%	10%	12%	18%	31%	56%	100%

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 fragrance*.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 contest 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 five scenarios, and later on you find a solution to the sixth. Your new submission must contain **all six output files**, since when you submit your new solutions the old ones will be thrown away.

Please email farior@farior.org if you have any questions about this procedure.