



# South African Programming Olympiad Day 1



## Overview

Problem	tomes	genghis	cunning
Source	tomes.java tomes.py tomes.c tomes.cpp tomes.pas	genghis.java genghis.py genghis.c genghis.cpp genghis.pas	cunning.java cunning.py cunning.c cunning.cpp cunning.pas
Input file	stdin	stdin	stdin
Output file	stdout	stdout	stdout
Time limit	1 second. Python: 10 seconds.	1 second. Python: 10 seconds.	2 seconds. Python: 20 seconds.
Memory limit	256MiB	256MiB	256MiB
Detailed feedback	No	No	20
<b>Total points</b>	<b>100</b>	<b>100</b>	<b>100</b>

The maximum total score is 300 points.

<http://olympiad.cs.uct.ac.za/contest.html>



# South African Programming Olympiad Day 1



## Reading Tomes

### Introduction

Sir Bedevere is in a pickle. After that disastrous affair with the Wooden Rabbit<sup>1</sup>, he suspects King Arthur no longer considers him the smartest knight in the land.

He has thus decided to do some reading in order to once again claim his place as the King's principle advisor in all matters not relating to sparrows.

### Task

Sir Bedevere has procured a number of large Tomes. Unfortunately, his reading speed is not very fast, and so he wants to know what pages to read in order to improve his knowledge of certain topics. Since the common index is yet to be invented, he has hired you to find pages that are relevant.

In order to fit as much knowledge on to as little paper as possible, the good monk who wrote the texts completely omitted any capital letters, punctuation or even spaces! All you have are  $L$  lines, and they might even break in the middle of a word!

Sir Belvedere has given you some lists of related words and the different ways in which they can be combined. These "combinations" are of the following form.

- Any sequence of lower case latin letters is a combination.
- Any sequence of lower case latin letters enclosed in braces ( { or } ) is a combination denoting some word from the list described in the curly braces.
- Any concatenation of combinations is a combination.

He wants you to count how many times the combinations appear in the page of text he is currently looking at. He doesn't care if the words overlap, and considers two occurrences identical only if they start at the same letter, and match the same combination.

### Example

Suppose Sir Bedevere is looking for content on heraldry. He knows two lists of useful words, colours and motifs. The colours are "Or" (gold), "Agent" (white), "Azure" (blue), "Gules" (red), "Sable" (black) and "Vert" (green). The motifs he remembers are "anchor", "grapes", "ribbon" and "annulets". Suppose the passage he is reading reads

<sup>1</sup>See *Monty Python and the Holy Grail*

"The Cape Town coat of arms is an erect, anchor Sable with a ribbon Azure and an escutchen Gules, charged with three annulets Or."

Naturally, this has had spaces, capitals and punctuation added in.

He knows that the proper way to refer to coloured object is to name the object and then the colour. He also remembers that the word "escutcheon" has something to do with heraldry. He is thus looking for snippets that look like {motif}{colour} or escutcheon{colour}

The first combination is matched three times, and the second once (shown in bold below).

"The Cape Town coat of arms is an erect, **anchor Sable** with a **ribbon Azure** and an **escutchen gules**, charged with three **annulets Or**."

### Input (stdin)

The first line of input contains three space separated integers,  $N$ ,  $M$  and  $L$ .

The next  $N$  lines describe a list that Sir Bedevere knows. The first word will be the name of the list. After a space will follow an integer indicating the number of words in the list,  $n_i$ . Then will follow that many space separated lower case words.

Thereafter are  $M$  lines, each describing a combination as described above.

Finally the last  $L$  lines will contain the text to search in. These will be all lower case latin letters with no spaces.

### Sample input

```
2 2 5
colour 6 or agent azure gules sable vert
motif 4 anchor grapes annulets ribbon
{motif}{colour}
escutcheon{colour}
thecapetowncoatofarmsisa
nerectanchorsablewithari
bbonazureandanescutcheon
guleschargedwiththreeann
uletsor
```

### Output (stdout)

The output should contain  $M$  lines. Each line contains a single integer: the number of occurrences of the corresponding combination in the input.



# South African Programming Olympiad Day 1



## Sample output

```
3  
1
```

## Constraints

- $1 \leq N \leq 50$
- $1 \leq n_i, M \leq 10$
- No combination exceeds 100 characters.
- $1 \leq L \leq 1000$
- The longest line of text does not exceed 100 characters.

Additionally, in 30% of the test cases,  $N = 0$ . In 10% of the test cases  $n_i = 1$  and in 10% of the test cases, each combination consists of a single word in curly braces.

## Time limit

1 second. Python: 10 seconds.

## Scoring

For each subtask, a correct solution will score 100% of the points available while an incorrect solution will score 0.



# South African Programming Olympiad Day 1



## The Stables of Genghis Khan

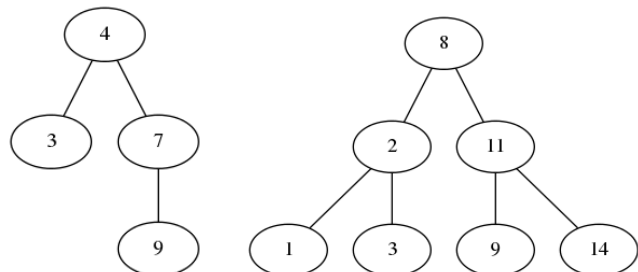
### Introduction

Bruce and Carl are having an excellent adventure... through time! They've been trying to teach people throughout history how to use algorithms, but no one has been listening. The Mongols, however, are the exception, and Genghis Khan has given them a chance to prove their worth. He has a number of horses, some of which are currently in his stables. He wants to find out which horses are in his stables and which aren't. Bruce and Carl are going to use binary search trees to help him do this quickly.

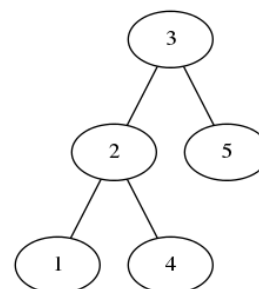
### Task

A *binary key tree* consists of a vertex (called the *root*) labelled with a number (called its *key*) and optionally up to two "subtrees" linked to it. These are called the left subtree and the right subtree and are also binary key trees.

A *binary search tree* is a binary key tree where the left and right subtrees (if they exist) are also binary search trees. Additionally, the key of the root is greater than every key in the left subtree and less than every key in the right subtree. For example, the following are binary search trees:



But the following aren't:



Genghis Khan has  $N$  horses, numbered 1 to  $N$ .  $K$  of them are currently in the stables, numbered  $X_1, X_2, \dots, X_K$ . Bruce and Carl want to make a binary search tree with vertices numbered with keys  $X_1, X_2, \dots, X_K$  in which they will quickly be able to look up if a horse is in the stable or not.

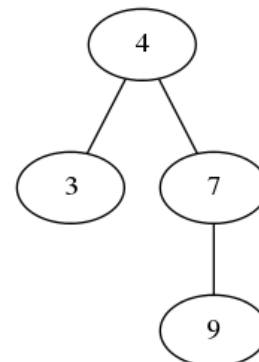
They use the following algorithm to search the tree for a key  $A$ :

1. Start at the root.
2. Compare the value of  $A$  to the key  $X$  of the current node. If  $A = X$  then they have found  $A$  and stop, if  $A < X$  they repeat step 2 on the left subtree. If  $A > X$  they repeat step 2 on the right subtree.
3. If at any stage the subtree they need to perform the algorithm on is not in the tree, then  $A$  is not in the tree and they stop.

The cost of looking for  $A$  is the number of times step 2 is performed during the execution of the algorithm. Help Bruce and Carl design a binary search tree which minimises the total cost of searching for all of the keys  $1, 2, \dots, N$ .

### Example

Suppose  $N = 10$ ,  $K = 4$  and the horses in the stable have numbers 3, 4, 7, 9. Then the optimal binary search tree has total cost 22, and is the following:



The costs of searching for each number are the following:



# South African Programming Olympiad Day 1



Number	Cost	Number	Cost
1	2	6	2
2	2	7	2
3	2	8	3
4	1	9	3
5	2	10	3

## Input (stdin)

The first line of input contains the integer  $N$ . The second line contains the integer  $K$ . The next  $K$  lines contain the integers  $X_1, X_2, \dots, X_K$ , one per line.

## Sample input

```
10
4
9
3
7
4
```

## Output (stdout)

Your program should output a single integer, the minimum total cost of a binary search tree satisfying the requirements.

## Sample output

```
22
```

## Constraints

- $1 \leq N \leq 10\,000\,000$
- $1 \leq K \leq 300$

Additionally, in 30% of the test cases,  $N, K \leq 10$

## Time limit

1 second. Python: 10 seconds.

## Scoring

A correct solution will score 100% while an incorrect solution will score 0%. A solution will be awarded partial marks if it solves some, but not all, test cases.



# South African Programming Olympiad Day 1



## Cunning Plan

### Introduction

Lord Blackmamba has a cunning plan. Many people bet money on a game called “Drop” at *Ye Aulde Faire*. Drop is played with a ball and a box which contains some obstacles. The ball is dropped into the box, it bounces off some obstacles, and players win money if they correctly predict where it drops out the bottom of the box. To stop people cheating, the game master also sometimes adds or removes obstacles from the box.

Blackmamba wants to predict the outcome of the game so he can scam the players out of their money. Help him figure out where the ball will drop out of the box if it is dropped into the box at a given position.

### Task

The box consists of  $M$  horizontal rows, numbered 1 to  $M$ , and  $N$  vertical columns numbered from 1 to  $N$ . The ball is dropped into the top of one of the columns. It continues to drop straight downwards in its current column until it hits an obstacle.

There are two kinds of obstacles. A ‘/’ obstacle knocks the ball one column left (after which it will continue to fall downwards until it hits another obstacle) and a ‘\’ obstacle knocks the ball one column right. Free space is represented by ‘.’

Additionally, sometimes the game master will change what is located in the  $i$ -th row and  $j$ -th column, either adding or removing an obstacle, or changing the type of obstacle there.

Your program will need to determine which column the ball will drop out of the bottom of the grid in, given which column it is dropped into at the top of the grid.

### Example

Suppose the grid is  $3 \times 3$  and is initially:

```
...  
./.  
...
```

If the ball is dropped into the third column, it will drop out the second. Now suppose the obstacle in the second row and third column is changed to ‘.’ If a ball is now dropped into the third column it will drop out the third column.

### Input (stdin)

The first line of input contains three space separated integers,  $M$ ,  $N$  and  $P$ . The following  $M$  lines each contain  $N$  characters. The  $i$ -th of these lines describes the  $i$ -th row of the box. Each of the next  $P$  lines is in one of two formats:

- “Q  $i$ ”, where  $i$  is an integer from 1 to  $N$ . For each of these lines, your program should output which column the ball drops out of if it is dropped into the  $i$ -th column.
- “U  $a$   $x$   $y$ ”, where  $a$  is “.”, “/” or “\”,  $x$  is an integer from 1 to  $M$  and  $y$  is an integer from 1 to  $N$ . This indicates that the item in the  $x$ -th row and  $y$ -th column of the box has been changed to  $a$ .

### Sample input

```
3 3 3  
...  
./.  
...  
Q 3  
U . 2 3  
Q 3
```

### Output (stdout)

For each line beginning with “Q”, your program should output a line containing a single integer, the index of the column which the ball will drop out of.

### Sample output

```
2  
3
```

### Constraints

- $1 \leq M, N \leq 1000$
- $1 \leq P \leq 100\,000$
- No obstacle will cause the ball to fall out of the left or right edge of the box.
- There will never be two obstacles in a “\” pattern.

Additionally,



# South African Programming Olympiad Day 1



Subtask	Points	Constraint
1	27	$1 \leq M, N \leq 20, 1 \leq P \leq 100$
2	12	There will be no lines in input beginning with "U".
3	7	There will be at most 10 lines in input beginning with "U".
4	54	There will be at most 10 000 lines in input beginning with "U".

## Time limit

2 seconds. Python: 20 seconds.

## Detailed feedback

Detailed feedback is enabled for this problem. You are limited to 20 submissions with detailed feedback.

## Scoring

For each subtask, a correct solution will score 100% of the points available while an incorrect solution will score 0.