Irish Collegiate Programming Competition 2012

Problem Set

University College Cork ACM Student Chapter

March 24, 2012

Contents

Instructions

Rules

- All mobile phones, laptops and other electronic devices must be powered off and stowed away for the duration of the contest.
- The only networked resource that teams are permitted to access is the submission system.
- If a team discovers an ambiguity or error in a problem statement they should tell an organiser. If the organisers agree that an ambiguity or error exists, a clarification will be issued to all teams.
- No multi-threading is allowed, and no sub processes.
- No file input/output is allowed. All input to your program will be provided via standard input (stdin) and all output should be printed to standard output (stdout).

Testing and Scoring

- When teams submit a solution the submission time will be recorded and the solution will be queued for testing.
- A problem will be tested against 10 test cases. All output from your program should be followed immediately by a new line and should not contain any additional whitespace.
- A solution will be accepted if it compiles on the test environment and produces the correct output for the sample inputs given in the question.
- If a solution is submitted while an earlier solution to the same problem is still in the queue, the earlier submission will not be tested. The earlier submission will be marked with the message: "Old, not going to be tested".
- All accepted problems will be scored out of 10 points, one for each test case that returns the correct output.
- You will not be told the points scored until the end of the contest.
- In the event of a tie on points, the winning team will be the one with the shortest amount of time between the contest start and their last solution that increased their score.

Submission Instructions

- Your password will be provided by the organisers. Please notify an organiser if you unable to log in.
- Submissions should consist of a single source file.
- To submit, click the "Submit a Solution" link, complete the submission form, upload your source file, and click save.
- Your submission should now be listed in your submission queue.
- Submission input is read from standard input and submission output should be written to standard output.
- To teams submitting Java solutions, be aware the testing script will be renaming your file to `Pn.java` (where `n` is the problem number) and as such the main class for problem 1 should be defined as:

```
public class P1 {

...

}
```

and will be called with:

```
java P1 < input-file
```

1 Equilibrium Index

An Equilibrium Index of a sequence of numbers is an index where the sum of all elements with a lower index is equal to the sum of all elements with a higher index (the number in the sequence at the index is not included in either sum).

In the sequence $\langle 1, 5, -7, 2, 3, -4, 0 \rangle$ the equilibrium indices are 3 and 6 since $1 + 5 + (-7) = 3 + (-4) + 0$ and $1 + 5 + (-7) + 2 + 3 + (-4) = 0$

Given a sequence of numbers, the task is to find the equilibrium indices of the sequence

Input

First line of input contains a single integer N, $0 < N < 10,000,000$. N is the length of the sequence. The following line contains n integers between $-1,000,000$ and $1,000,000$ inclusive, separated by spaces and followed by a newline character.

Output

The output should be a list of space separated equilibrium indices for the given list sorted from lowest to highest index.

Sample Input 1

```
4
6 2 5 1
```

Sample Output 1

```
1
```

Sample Input 2

```
7
1 5 -7 2 3 -4 0
```

Sample Output 2

```
3 6
```

2 Flipping Numbers

Given a list of N unique integers, $2 \leq N \leq 25000$, produce a sequence of "flips" so that the end result is the list, sorted in ascending order. A k-flip takes the first k numbers and reverses their order in the list. For example given the list $\langle 2, 6, 4, 1, 9 \rangle$, a k-flip where $k = 4$ would result in the list $\langle 1, 4, 6, 2, 9 \rangle$.

Your task is to determine a list of k values which represent the sequence of flips that will result in list being sorted.

Input

The first line of input contains the integer N which represents the number of integers in the list. The next line of input contains N space-separated integers.

Output

Your program should output the sequence of k values which represent the flips that turn the input list in to a sorted list. Each of these values should be separated by a space and the output should be terminated by a new line character.

Sample Input 1

```
5
1 4 6 2 9
```

Sample Output 1

```
3 4 2
```

3 Cruise Ship

A cruise ship runs tours which visit a number of ports. On a tour, the cruise ship will visit each of the ports at least once and may visit a port multiple times. Traveling between ports requires the cruise ship to burn fuel. The sea leanes between ports vary in length, and as such some sea leanes require more fuel than others.

The ship can only carry a certain amount fuel in its fuel tank. However, when the ship stops at a port it refills its tank completely. The task is to find the smallest possible fuel tank which allows the ship to complete a tour. It must be noted that the ship may stop at other ports to refuel if it wants to get between two ports which are a large distance apart.

Ships are limited to traveling in predefined lanes, therefore there may not be a direct route between every two ports. However, every port is reachable. You are given the list of all sea lanes.

Input

The first line of the input contains two integers N and M, separated by spaces. N is the number of ports, $2 \leq N \leq 10000$ and M is the number of possible routes, $1 \leq M \leq N^2$. The rest of the input describes the possible sea lanes between ports. Each of the following M lines contain 3 integers a, b and f, separated by spaces. a and b are two ports, $0 \leq a, b < N$, and f is the fuel cost for the lane.

Output

An integer, the size of the smallest necessary fuel tank, followed by a new line character.

Sample Input 1

```
4 6
1 3 6
1 2 4
4 2 3
3 4 5
2 3 7
1 4 2
```

Sample Output 1

```
5
```

Sample Input 2

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

Sample Output 2

```
3
```

4 Postfix Assignment

Given a list of variables, a mathematical expression and a list of values, find an assignment of the values to the variables that results in the expression being evaluated to true.

A postfix expression is an expression in which the operators occur in the expression after their operands. An operator always takes two operands. An operand can be either a variable or another operator/sub-expression.

An example of a postfix expression would be $3\ 5\ \times$. This is equivalent to 3×5, resulting in an answer of 15. $6\ 4\ \times\ 2\ +$ would be evaluated as $2 + (6 \times 4)$, resulting in an answer of 26.

Input

The first line contains an integer N, $2 \leq N \leq 26$, the number of variables in the expression.

The second line contains N space separated single character variable names that are used in the expression.

The third line contains N space separated integers $0 \leq x_i \leq 100$ that can be assigned to the variables. There will be exactly one value for each variable.

The fourth line will contain a postfix expression containing variables from the previously defined list and a combination of the operators $*\ +\ -$. There will be no division operator. Each variable name or operator is separated by a space and will always be one character long. Variables may occur multiple times in an expression. The last operator will always be an equality comparison '$=$' operator and it will be the only one in the equation.

Output

The output of your program should be a space separated list of values to be assigned to the variables, in the order the variables were given. The first value should be the smallest possible value that could be assigned to the first variable, the second the smallest possible to be assigned to the second given the first's value and so on.

Sample Input 1

```
3
a b c
2 5 8
a c * b - c =
```

Sample Output 1

```
5 8 2
```

Sample Input 2

```
2
b a
8 4
a b - a b - b - + b =
```

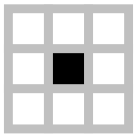
Sample Output 2

```
4 8
```

In *Sample Input 2* the expression can be written as $(a \times c) - b = c$. In the output for this input we assign 5 to a, 8 to b, and 2 to c.

5 Mending Still Lives

Conway's Game of Life is played on a $N * M$ grid. Each square of the grid is known as a cell, which at any time during the game is either dead or alive. A cell has eight neighbours as shown below.



The eights neighbors surrounding a cell.

The arrangement of living and dead cells at time t leads to a new arrangement at time $t+1$ according to the following rules:

- if a cell has exactly three living neighbours at time t, it is alive at time $t+1$.
- if a cell has exactly two living neighbours at time t it is in the same state at time $t+1$ as it was at time t.
- otherwise, the cell is dead at time $t+1$

A still-life is an arrangement that does not change between times t and $t+1$. Given an initial state of an $N \times M$ grid and a cell limit k, find the k, or less, alive cells to added to make the arrangement a still-life.

Input

The first line will contain 3 integers, N, M and k, $2 \leq N, M \leq 50$ and $1 \leq k \leq 10$. N and M represent the height and width of the grid respectively. k is the maximum number of alive cells which can be added. The following N lines will contain m space separated binary cell states, 1 for alive and 0 otherwise.

Output

The first line of output must contain an integer s, which is the number of alive cells to be added to make the arrangement a still life. The following s lines should contains the row and column coordinates of those cells (zero indexed from top left as in the input).

Sample Input 1

```
4 4 1
0 0 0 0
0 1 1 0
0 1 0 0
0 0 0 0
```

Sample Output 1

```
1
2 2
```

Sample Input 2

```
4 6 1
0 0 0 0 0 0
0 1 1 0 1 0
0 1 0 0 1 0
0 0 0 0 0 0
```

Sample Output 2

```
1
2 3
```

6 Numerical Maze

Given a grid of cells with numerical values you are tasked with finding the shortest path from a position on the top row down to the bottom row that matches the following sequence: $1; 1, 2; 1, 2, 3; 1, 2, 3, 4; \ldots$ Each consecutive step must be either a step up, down, left or right of the current cell, diagonal movement is not allowed.

You are tasked with finding the leftmost entry point and exit point. You should start on a cell with value 1 and the exit point must be the last digit in the current sub sequence, for example you *can't* have a path as follows $1; 1, 2; 1, 2$ as the last sequence should have finished on a 3.

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

An example maze

Input

The first line of input contains two integers, $1 \leq N, M \leq 50$. These designate the number rows and columns of the grid respectively. The next N lines will contain M space separated integers; the values of the grid cells.

Output

The first line of output must contain the coordinates (zero indexed, from the top left) for the entry point followed by a newline character. Following this should be the coordinates of the exit point.

If there are several solutions, print the one with lexicographically smallest starting point. If there is still a tie, print the one with lexicographically smallest ending point.

Sample Input 1

```
2 2
1 1
3 2
```

Sample Output 1

```
0 0
1 1
```

Sample Input 2

```
10 11
1 6 5 2 1 1 2 3 2 1 4
1 2 6 3 2 1 1 3 4 5 6
1 2 3 2 1 3 2 5 6 4 2
2 3 1 2 2 3 3 4 5 2 1
3 4 2 3 4 5 3 2 1 4 2
4 3 4 4 5 6 4 3 2 5 3
5 4 2 1 2 3 4 4 3 6 4
6 5 3 2 3 4 5 5 4 1 1
1 6 4 3 5 5 6 6 1 2 3
2 1 5 1 6 6 1 2 2 3 4
```

Sample Output 2

```
0 5
9 2
```

7 Word Counting

Given a string s of length N, $0 < N \leq 2500000$, and a dictionary of valid words of size M, $0 < M \leq 80000$, the goal is to count the total number of words that can be found in the string.

The string s, and the words in the dictionary may contain a mixture of upper and lower case. However, for the purpose of counting word occurences, a valid match based on characters, not case (i.e. matching is case insensitive).

Input

First line contains two integers, N and M where N is the size of the string and M is the size of the dictionary. The second line contains a string followed by a newline. This is the string which to search for word occurrences. The remaining input describes the dictionary. The each line next M lines contain a string followed by a newline. Each of these strings is a valid word in the dictionary.

Output

The output should be a single integer followed by newline. This should be the count of word occurrences in the input string.

Sample Input 1

```
7 3
saddeet
add
for
set
```

Sample Output 1

```
1
```

Sample Input 2

```
11 6
taretesarts
art
arts
tar
tart
tarts
tsar
```

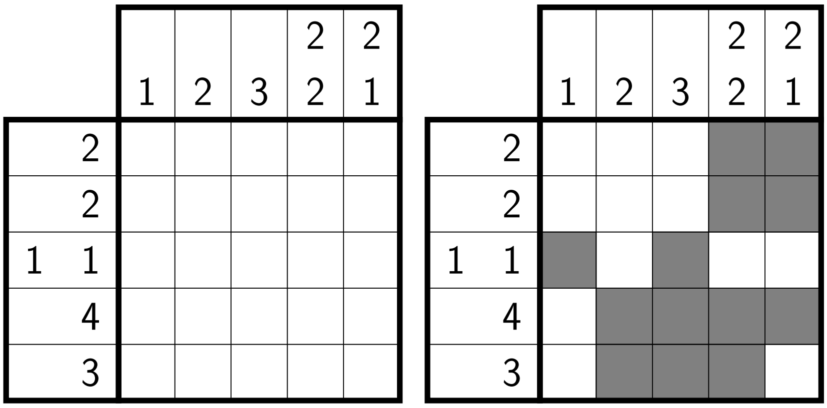
Sample Output 2

```
3
```

8 Nonograms

Nonograms are a logic puzzle played on an $N \times M$ grid. The aim is to fill cells in the grid in a particular pattern. Each row and column of the grid has a series of clues associated with it, which describes the filled in cells. These clues are in the form of a series of integers, $C = \langle c_0, c_1, ..., c_l \rangle$, where each element of C is a a group of filled in cells, separated by blank cells. In C, c_0 is the first group, c_1 is the second group, and so on. For each row, the first group is the left most group, and for each column, the first group is the top most group.

Given a grid, the task is to find the unique solution for that grid.



(a) Empty puzzle (b) Completed puzzle

5×5 Nonogram

Input

The first line contains two integers, N and M, $5 \leq N, M \leq 25$, followed by a newline. N and M are the height and width of the board respectively.

The following N lines of input contain the clues for each row. The remaining M lines contains the clues for each columns. These $N + M$ lines contain a sequence of space separated integer values, followed by a new line. On each of these lines the first integer is the number of clues, and the remain integers are the clues. If there are no clues for a row or column, then the line will contain the integer 0.

Output

The task is to output the completed grid. The output should contain N lines of length M containing ones and zeros, representing the filled in cells. 1 indicates a filled cell and 0 indicates an empty cell.

Sample Input 1

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

Sample Output 1

```
00011
00011
10100
01111
01110
```

Sample Input 2

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

Sample Output 2

```
10001
10110
10110
01101
10011
10010
00111
11111
11110
10100
```