acm International Collegiate Programming Contest

IBM event sponsor

2010 ACM ICPC Southeast USA Regional Programming Contest

6 November, 2010

PROBLEMS

Hosted by:

Florida Institute of Technology
Armstrong Atlantic State University
University of West Florida

acm International Collegiate Programming Contest

IBM event sponsor

A: Balloons

As you may know, balloons are handed out during ACM contests to teams as they solve problems. However, this sometimes presents logistical challenges. In particular, one contest hosting site maintains two rooms, A and B, each containing a supply of balloons. There are **N** teams attending the contest at that site, each sitting at a different location. Some are closer to room A, others are closer to room B, and others are equally distant. Given the number of balloons needed by each team and the distance from each team to room A, and to room B, what is the minimum total possible distance that must be traveled by all balloons as they are delivered to their respective teams, assuming they are allocated in an optimal fashion from rooms A and B? For the purposes of this problem, assume that all of the balloons are identical.

The Input

There will be several test cases in the input. Each test case will begin with a line with three integers:

```
N A B
```

Where **N** is the number of teams (1 ≤ **N** ≤ 1,000), and **A** and **B** are the number of balloons in rooms A and B, respectively (0 ≤ **A**,**B** ≤ 10,000). On each of the next N lines there will be three integers, representing information for each team:

```
K DA DB
```

Where **K** is the total number of balloons that this team will need, **DA** is the distance of this team from room A, and **DB** is this team's distance from room B (0 ≤ **DA**,**DB** ≤ 1,000). You may assume that there are enough balloons – that is, Σ(**K**'s) ≤ **A**+**B**. The input will end with a line with three 0s.

The Output

For each test case, output a single integer, representing the minimum total distance that must be traveled to deliver all of the balloons. Count only the outbound trip, from room A or room B to the team. Don't count the distance that a runner must travel to return to room A or room B. Print each integer on its own line with no spaces. Do not print any blank lines between answers.

Sample Input

```
3 15 35
10 20 10
10 10 30
10 40 10
0 0 0
```

Sample Output

```
300
```

B: Bit Counting

Start with an integer, N_0, which is greater than 0. Let N_1 be the number of ones in the binary representation of N_0. So, if N_0=27, N_1=4. For all i>0, let N_i be the number of ones in the binary representation of N_{i-1}. This sequence will always converge to one. For any starting number, N_0, let K be the minimum value of i>=0 for which N_i=1. For example, if N_0=31, then N_1=5, N_2=2, N_3=1, so K=3.

Given a range of consecutive numbers, and a value X, how many numbers in the range have a K value equal to X?

Input

There will be several test cases in the input. Each test case will consist of three integers on a single line:

```
LO HI X
```

Where **LO** and **HI** ($1 \le$ **LO** $\le$ **HI** $\le 10^{18}$) are the lower and upper limits of a range of integers, and **X** ($0 \le$ **X** ≤ 10) is the target value for K. The input will end with a line with three 0s.

Output

For each test case, output a single integer, representing the number of integers in the range from **LO** to **HI** (inclusive) which have a K value equal to **X** in the input. Print each integer on its own line with no spaces. Do not print any blank lines between answers.

Sample Input

```
31 31 3
31 31 1
27 31 1
27 31 2
1023 1025 1
1023 1025 2
0 0 0
```

Sample Output

```
1
0
0
3
1
1
```

C: Data Recovery

James obtained some very important data for his company and wrote them down in a table. Each cell of this table contains an integer between 0 and 100 inclusive. For the purpose of collecting statistics, James also wrote down the sum of every row, and the sum of every column of the table. Now he only needs to show this to his boss and he will surely get a promotion.

Unfortunately, on his way to the executive meeting, there was heavy rain. The piece of paper with table data is partially wet and some cells become unreadable! Luckily, the statistics he collected are all intact.

Not wanting to miss this opportunity for a promotion, he is wondering whether he can recover his lost data using the statistical information he collected. That is, with the additional knowledge of the sum of every row and column, maybe he can recover the exact value inside each of the now unreadable cells. He has asked you to accomplish this task for him.

Input

There will be several test cases in the input. Each test case will begin with one line containing two integers **N** and **M** ($1 \le \mathbf{N},\mathbf{M} \le 50$), the dimension of the table. Then, the next **N** lines will each contain **M** integers. Each integer will be either between 0 and 100 inclusive, indicating a table cell that is still legible, or the value -1, indicating the cell is now illegible because of the rain. After the **N** lines containing table entries, there will be two more lines. The first line will contain **N** integers, each between 0 and 5,000 inclusive, indicating the sum of each row in the table, from topmost row to the bottommost row. The second line will contain **M** integers, each between 0 and 5,000 inclusive, indicating the sum of each column in the table, from the leftmost column to the rightmost column. The input will end with a line which contains two 0s.

You may assume that the given table is valid. That is, there exists a way to replace every -1 in the input table with an integer between 0 and 100 inclusive, such that sum of every row and column of the table equals the sums given in the input.

Output

For each test case, print **N** lines with **M** entries each. Each entry must either be an integer between 0 and 100 inclusive, or -1. If the table cell's value in the input is between 0 and 100 inclusive, then this value should be printed to the output. If the table cell's value is -1 in the input, but among all integers between 0 and 100 inclusive, only one can be put in this cell and be consistent with the known values and statistics, then output this unique integer. Otherwise, if the value of the cell cannot be uniquely determined, print a -1 for this cell in the output. Print a single space between consecutive entries on one line, without trailing spaces on any line. Do not print any blank lines between outputs.

Sample Input

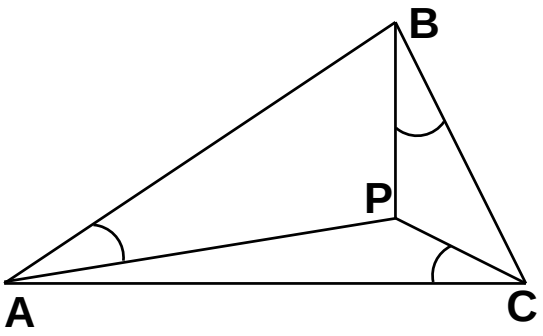
```
2 2
1 -1
-1 -1
100 100
100 100
5 5
-1 -1 6 -1 8
-1 -1 0 4 2
3 -1 -1 5 -1
4 0 2 -1 -1
2 1 5 -1 -1
21 10 15 13 14
18 6 17 15 17
2 3
1 2 3
3 5 6
6 14
4 7 9
0 0
```

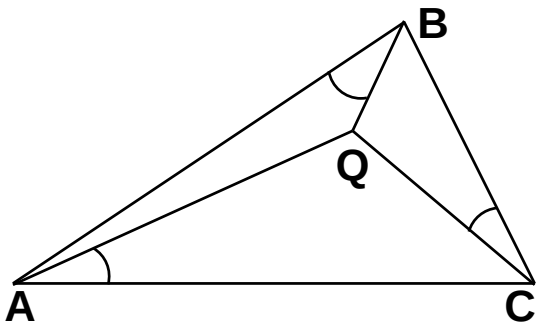
Sample Output

```
1 99
99 1
-1 -1 6 0 8
-1 -1 0 4 2
3 3 4 5 0
4 0 2 -1 -1
2 1 5 -1 -1
1 2 3
3 5 6
```

D: Equal Angles

The All-Equal company has been tasked with placing towers on triangular plots so that the angles formed between the towers and the sides of the plots are equal. Given a triangle defined by points A, B and C, there are two such points - call them P and Q. There is one where angles PAB = PBC = PCA, and one where angles QBA = QCB = QAC.





Input

There will be several test cases in the input. Each test case will consist of six integers on a single line:

AX AY BX BY CX CY

Each integer will be in the range from -100 to 100. These integers represent the three points of the triangle: (**AX**,**AY**), (**BX**,**BY**) and (**CX**,**CY**). The points are guaranteed to form a triangle: they will be distinct, and will not all lie on the same line. The input will end with a line with six 0s.

Output

For each test case, output four space-separated real numbers:

PX PY QX QY

Where (**PX**,**PY**) and (**QX**,**QY**) are the requested points. Print each real number with exactly two decimal places, rounded, and put a single space between them. Print no blank lines between outputs.

Sample Input

```
-4 5 -10 0 7 0
-15 -5 15 -5 0 21
0 0 0 0 0 0
```

Sample Output

```
-6.26 1.28 -1.96 3.07
0.01 3.66 -0.01 3.66
```

E: Maximum Square

Given an NxM matrix of all 1s and 0s, find the largest submatrix which is a square containing all 1s.

Input

There will be several test cases in the input. Each test case will begin with two integers, **N** and **M** (1 ≤ **N**,**M** ≤ 1,000) indicating the number of rows and columns of the matrix. The next N lines will each contain M space-separated integers, guaranteed to be either 0 or 1. The input will end with a line with two 0s.

Output

For each test case, print a single integer, indicating the width (and height) of the largest square of all 1s, or 0 if there are no 1s. Print no extra spaces, and do not print any blank lines between answers.

Sample Input

```
4 5
0 1 0 1 1
1 1 1 1 1
0 1 1 1 0
1 1 1 1 1
3 4
1 1 1 1
1 1 1 1
1 1 1 1
6 6
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0
```

Sample Output

```
3
3
0
```

F: Palindrometer

While driving the other day, John looked down at his odometer, and it read 100000. John was pretty excited about that. But, just one mile further, the odometer read 100001, and John was REALLY excited! You see, John loves palindromes – things that read the same way forwards and backwards. So, given any odometer reading, what is the least number of miles John must drive before the odometer reading is a palindrome? For John, every odometer digit counts. If the odometer reading was 000121, he wouldn't consider that a palindrome.

The Input

There will be several test cases in the input. Each test case will consist of an odometer reading on its own line. Each odometer reading will be from 2 to 9 digits long. The odometer in question has the number of digits given in the input - so, if the input is 00456, the odometer has 5 digits. There will be no spaces in the input, and no blank lines between input sets. The input will end with a line with a single 0.

The Output

For each test case, output the minimum number of miles John must drive before the odometer reading is a palindrome. This may be 0 if the number is already a palindrome. Output each integer on its own line, with no extra spaces and no blank lines between outputs.

Sample Input

```
100000
100001
000121
00456
0
```

Sample Output

```
1
0
979
44
```

G: Profits

Your friends have just opened a new business, and you want to see how well they are doing. The business has been running for a number of days, and your friends have recorded their net profit on each day. You want to find the largest total profit that your friends have made during any consecutive time span of at least one day. For example, if your friends' profits looked like this:

```
Day 1: -3
Day 2: 4
Day 3: 9
Day 4: -2
Day 5: -5
Day 6: 8
```

Their maximum profit over any span would be 14, from days 2 to 6.

Input

There will be several test cases in the input. Each test case will begin with an integer **N** (1 ≤ **N** ≤ 250,000) on its own line, indicating the number of days. On each of the next **N** lines will be a single integer **P** (-100 ≤ **P** ≤ 100), indicating the profit for that day. The days are specified in order. The input will end with a line with a single 0.

Output

For each test case, output a single integer, representing the maximum profit over any non-empty span of time. Print each integer on its own line with no spaces. Do not print any blank lines between answers.

Sample Input

```
6
-3
4
9
-2
-5
8
2
-1000
-19
0
```

Sample Output

```
14
-19
```

H: Roller Coaster

Bessie has gone on a trip, and she's riding a roller coaster! Bessie really likes riding the roller coaster, but unfortunately she often gets dizzy.

The roller coaster has a number of distinct sections that Bessie rides in order. At the beginning of the ride, Bessie's dizziness and fun levels are both at 0. For each section of the roller coaster, Bessie can either keep her eyes open or keep them closed (and must keep them that way for the whole section). If she keeps her eyes open for a section, her total fun increases by a Fun factor for that section, and her dizziness increases by a Dizziness factor for that section. However, if she keeps her eyes closed for the section, her total fun will not change, but her dizziness will decrease by a value that's constant for the entire roller coaster. (Note that her dizziness can never go below 0.)

If, at any point, Bessie's dizziness is above a certain limit, Bessie will get sick. Write a program to find the maximum amount of fun Bessie can have without getting sick.

Input

There will be several test cases in the input. Each test case will begin with a line with three integers:

```
N K L
```

Where **N** ($1 \le N \le 1{,}000$) is the number of sections in this particular roller coaster, **K** ($1 \le K \le 500$) is the amount that Bessie's dizziness level will go down if she keeps her eyes closed on any section of the ride, and **L** ($1 \le L \le 300{,}000$) is the limit of dizziness that Bessie can tolerate – if her dizziness ever becomes larger than **L**, Bessie will get sick, and that's not fun!

Each of the next N lines will describe a section of the roller coaster, and will have two integers:

```
F D
```

Where **F** ($1 \le F \le 20$) is the increase to Bessie's total fun that she'll get if she keeps her eyes open on that section, and **D** ($1 \le D \le 500$) is the increase to her dizziness level if she keeps her eyes open on that section. The sections will be listed in order. The input will end with a line with three 0s.

Output

For each test case, output a single integer, representing the maximum amount of fun Bessie can have on that roller coaster without exceeding her dizziness limit. Print each integer on its own line with no spaces. Do not print any blank lines between answers.

Sample Input

```
3 1 2
2 1
3 1
5 2
10 5 1
20 2
12 4
3 3
10 6
20 3
19 9
19 7
1 500
15 5
4 2
0 0 0
```

Sample Output

```
7
0
```

I: Skyline

The director of a new movie needs to create a scaled set for the movie. In the set there will be **N** skyscrapers, with distinct integer heights from 1 to **N** meters. The skyline will be determined by the sequence of the heights of the skyscrapers from left to right. It will be a permutation of the integers from 1 to **N**.

The director is extremely meticulous, so she wants to avoid a certain sloping pattern. She doesn't want for there to be ANY three buildings in positions i, j and k, i < j < k, where the height of building i is smaller than that of building j, and building j's height is smaller than building k's height.

Your task is to tell the director, for a given number of buildings, how many distinct orderings for the skyline avoid the sloping pattern she doesn't like.

Input

There will be several test cases in the input. Each test case will consist of a single line containing a single integer **N** (3 ≤ **N** ≤ 1,000), which represents the number of skyscrapers. The heights of the skyscrapers are assumed to be 1, 2, 3, …, N. The input will end with a line with a single 0.

Output

For each test case, output a single integer, representing the number of good skylines - those avoid the sloping pattern that the director dislikes - **modulo 1,000,000**. Print each integer on its own line with no spaces. Do not print any blank lines between answers.

Sample Input

```
3
4
0
```

Sample Output

```
5
14
```

J: Underground Cables

A city wants to get rid of their unsightly power poles by moving their power cables underground. They have a list of points that all need to be connected, but they have some limitations. Their tunneling equipment can only move in straight lines between points. They only have room for one underground cable at any location except at the given points, so no two cables can cross.

Given a list of points, what is the least amount of cable necessary to make sure that every pair of points is connected, either directly, or indirectly through other points?

Input

There will be several test cases in the input. Each test case will begin with an integer **N** ($2 \le N \le 1{,}000$), which is the number of points in the city. On each of the next **N** lines will be two integers, **X** and **Y** ($-1{,}000 \le X,Y \le 1{,}000$), which are the (**X**,**Y**) locations of the **N** points. Within a test case, all points will be distinct. The input will end with a line with a single 0.

Output

For each test case, output a single real number, representing the least amount of cable the city will need to connect all of its points. Print this number with exactly two decimal places, rounded. Print each number on its own line with no spaces. Do not print any blank lines between answers.

Sample Input

```
4
0 0
0 10
10 0
10 10
2
0 0
10 10
0
```

Sample Output

```
30.00
14.14
```