1996-1997 Asia Regional
ACM International Collegiate Programming Contest
Shanghai University, Shanghai, P. R. China, Nov. 3, 1996

Problem A

Schedule Problem

Input file: schedule.in

A project can be divided into several parts. Each part should be completed continuously. This means if a part should take 3 days, we should use a continuous 3 days do complete it. There are four types of constrains among these parts which are FAS, FAF, SAF and SAS. A constrain between parts is FAS if the second one should finish after the first one started. FAF is finish after finish. SAF is start after finish, and SAS is start after start. Assume there are enough people involved in the projects, which means we can do any number of parts concurrently. You are to write a program to give a schedule of a given project, which has the shortest time.

Input

The input file consists a sequences of projects, with an empty line indicates the end of input.

Each project consists the following lines:
the count number of parts (one line)
times should be taken to complete these parts, each time occupies one line
a list of FAS, FAF, SAF or SAS and two part number indicates a constrain of the two parts
a line only contains a '#' indicates the end of a project

Output

Output should be a list of lines, each line includes a part number and the time it should start. Time should be a non-negative integer, and the start time of first part should be 0. If there is no answer for the problem, you should give a non-line output containing "impossible".

A blank line should appear following the output for each project.

Sample Input

```
3
2
3
4
SAF 1 2
FAF 2 3
#
3
1
1
1
SAF 1 2
SAF 2 3
SAF 3 1
#
```

Output for the Sample Input

```
Case 1:
1 0
2 2
3 1

Case 2:
impossible
```

1996-1997 Asia Regional

ACM International Collegiate Programming Contest

Shanghai University, Shanghai, P. R. China, Nov. 3, 1996

Problem B

Minimum Transport Cost

Input file: trans.in

These are N cities in Spring country. Between each pair of cities there may be one transportation track or none. Now there is some cargo that should be delivered from one city to another. The transportation fee consists of two parts:

1. The cost of the transportation on the path between these cities, and
2. a certain tax which will be charged whenever any cargo passing through one city, except for the source and the destination cities.

You must write a program to find the route which has the minimum cost.

Input

The data of path cost, city tax, source and destination cities are given in the file CITY.IN, which is of the form:

a_{11} a_{12} …… a_{1N}

a_{21} a_{22} …… a_{2N}

………………

a_{N1} a_{N2} …… a_{NN}

b_1 b_2 …… b_N

c d

e f

………

g h

where a_{ij} is the transport cost from city i to city j, a_{ij} = -1 indicates there is no direct path between city i and city j. b_i represents the tax of passing through city i. And the cargo is to be delivered from city c to city d, city e to city f, ……, and city g to city h. You must output the sequence of cities passed by and the total cost which is of the form:

Output

From c to d:

path: c-->c_1-->……-->c_k-->d

Total cost: ……

…

From e to f:
path: e-->e_1-->………-->e_k-->f
Total cost: ……

Sample Input

```
0  3  22  -1  4
3  0  5  -1  -1
22  5  0  9  20
-1  -1  9  0  4
4  -1  20  4  0
5  17  8  3  1
1  3
3  5
2  4
```

Output for the Sample Input

```
From 1 to 3 :
Path: 1-->5-->4-->3
Total cost : 21

From 3 to 5 :
Path: 3-->4-->5
Total cost : 16

From 2 to 4 :
Path: 2-->1-->5-->4
Total cost : 17
```

1996-1997 Asia Regional
ACM International Collegiate Programming Contest
Shanghai University, Shanghai, P. R. China, Nov. 3, 1996

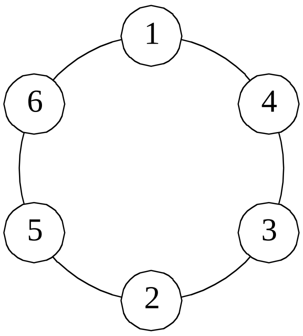
Problem C

Prime Ring Problem

Input file: prime.in

A ring is compose of *n* (even number) circles as shown in diagram. Put natural number 1, 2, …, *n* into each circle separately, and the sum of numbers in two adjacent circles should be a prime.

Note: the number of first circle should always be 1.



Input

n ($0 < n \leq 20$)

Output

The output format is shown as sample below. Each row represents a series of circle numbers in the ring beginning from 1 clockwisely and anticlockwisely. The order of numbers must satisfy the above requirements.

You are to write a program that completes above process.

Sample Input

```
6
8
```

Output for the Sample Input

```
Case 1:
1 4 3 2 5 6
1 6 5 2 3 4

Case 2:
1 2 3 8 5 6 7 4
1 2 5 8 3 4 7 6
1 4 7 6 5 8 3 2
1 6 7 4 3 8 5 2
```

1996-1997 Asia Regional
ACM International Collegiate Programming Contest
Shanghai University, Shanghai, P. R. China, Nov. 3, 1996

Problem D

Milk Bottle Data

Input file: bottle.in

There is a box of the shape of an N*M lattice. Each grid of the lattice may contain a milk bottle or none. Mr. Smith wrote down the data of the box by making a record for each row from left to right and each column from top to bottom. In each record, ‘1’ indicates that there is a bottle in the corresponding grid and ‘0’ does not. Unfortunately, the order of these records is thrown into confusion, and some of these records have corrupted.

Now it’s up to you to provide a program to recover these data: i.e. to give the original arrangement of the box and give real values for those corrupted data.

Input

The input data is stored in file BOTTLE.IN, where ‘2’ denotes that the corresponding character has been corrupted. Each line in the file represents a record. You should output the original arrangement of the box, and show the record number for each column on the top of the lattice, and show the record number for each row to the left of the lattice.

Output

You are required to give only one possible result if there are many, and you should give an indication if there is no possibility of original arrangement.

Sample Input

```
01210
21120
21001
12110
12101
12101
00011
22222
11001
10010
```

Output for the Sample Input

```
   9  8  6  2  7
4  1  0  1  1  0
10 1  0  0  1  0
1  0  1  1  1  0
3  0  1  0  0  1
5  1  1  1  0  1
```

1996-1997 Asia Regional
ACM International Collegiate Programming Contest
Shanghai University, Shanghai, P. R. China, Nov. 3, 1996

Problem E

String Distance and Transform Process

Input file: string.in

String Distance is a non-negative integer that measures the distance between two strings. Here we give the definition. A transform list is a list of string, where each string, except for the last one, can be changed to the string followed by adding a character, deleting a character or replacing a character. The length of a transform list is the count of strings minus 1 (that is the count of operations to transform these two strings). The distance between two strings is the length of a transform list from one string to the other with the minimal length. You are to write a program to calculate the distance between two strings and give the corresponding transform list.

Input

Input consists a sequence of string pairs, each string pair consists two lines, each string occupies one line. The length of each string will be no more than 80.

Output

For each string pair, you should give an integer to indicate the length between them at the first line, and give a sequence of command to transform string 1 to string 2. Each command is a line lead by command count, then the command. A command must be

Insert pos, value
Delete pos
Replace pos, value

where pos is the position of the string and pos should be between 1 and the current length of the string (in Insert command, pos can be 1 greater than the length), and value is a character. Actually many command lists can satisfy the request, but only one of them is required.

Sample Input

```
abcac
bcd
aaa
aabaaaa
```

Output for the Sample Input

```
3
1 Delete 1
2 Replace 3,d
3 Delete 4
4
1 Insert 1,a
2 Insert 2,a
3 Insert 3,b
4 Insert 7,a
```

1996-1997 Asia Regional
ACM International Collegiate Programming Contest
Shanghai University, Shanghai, P. R. China, Nov. 3, 1996

Problem F

The Partition of a Cake

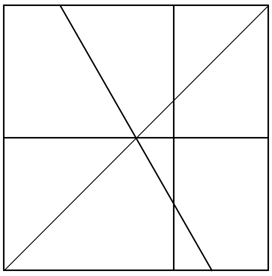
Input file: partcake.in

There is a 1000*1000 square cake. We use knife to cut the cake. The problem is after a series of cutting, how many partitions the cake will has.

Assumption:

1. The number of the cutting will be no more than 8.
2. After the cutting, the length of any edge of the partition will no less than 1.
3. The vertex coordinates of the cake are (0, 0)(0, 1000)(1000, 1000)(1000, 0).
4. The intersections of the cut line and the cake edge are two.

The following Graph is a sample partition. The number of the partitions is 10.



Input

The first line of the input is the number of the cutting. The following lines contain the information of the cut lines. Each line has 4 integer number, which represent the coordinate of the intersection of the line and the cake edge.

Output

The output is the number of the partitions of the cake.

Sample Input

```
3
0 0 1000 1000
500 0 500 1000
0 500 1000 500
```

Output for the Sample Input

```
6
```

1996-1997 Asia Regional
ACM International Collegiate Programming Contest
Shanghai University, Shanghai, P. R. China, Nov. 3, 1996

Problem G

Input file: .in

The Problem of Train Setout

Input file: train.in

At a railway station there is only one channel. There are two kinds of train: the express train and ordinary train. The ordinary train should wait to setout until all the express train have set-out. But if an ordinary train is just setouting, the incoming express train should wait until the ordinary train leaves. The leaving time of express train and ordinary train are Le and Lo. And the time between the arriving of two express train is Ae and of ordinary train is Ao. The total observation time is T.

You should work out a program to illustrate the number of trains waiting to setout at any time, and calculate the average time of the two kinds of train waiting We, Wo, and the average time of the all trains waiting Wt, and the time every train waited W(t).

Assume that the setout channel is empty at first and a express train comes first. And the output channel will never be empty.

Input

The input is a line of five numbers that represent Ae, Ao, Le, Lo, T respectively.

Output

The first part of the output is an array of the waiting trains, one element per line. There are three numbers in a line. First is the time, second is the number of waiting express trains, the third is the number of waiting ordinary trains. At the end of the array, there is a 0.

After the array, there is a line contains three numbers are We, Wo, Wt.

The last part of the output is also an array, one element per line. There are two elements per line. The first one is the train. If the train is an express train, then the first element begins with **E**, followed by the number of the express train, (the number is according to the incoming order of the train); else it begins with **O**, followed by the number of the ordinary train. The second is the time the train has waited.

Sample Input

```
3 6 6 2 30
```

Output for the Sample Input

```
0       0   0
3       1   0
6       1   1
9       2   1
12      2   2
15      3   2
18      3   3
21      4   3
24      4   4
27      5   4
30      5   5
0
6 12 8
0
E1 0
E2 3
E3 6
E4 9
E5 12
E6
15
E7 12
E8 9
E9 6
E10 3
E11 0
O1 24
O2 18
O3 12
O4 6
O5 0uters
```