National Collegiate Programming Contest 1998

Notes for Contestants

1. There are 6 problems for the contest. You may work on the problems in any order.
2. You may use any algorithm/method to solve the problems. However, the execution time of your program for each problem must not exceed 30 seconds, otherwise the program will be considered **run-time exceeded**.
3. The judging system, **PC^2**, is provided by ACM. For detailed information of system environment and operational instructions, please refer to the additional *Contest Handbook*.
4. Input files for the problems are preinstalled in the system for automatic judging. The input files are named according to the problem ID, where "**px.dat**" is the input file for **Problem X**.
5. If you have any questions during the contest, you may communicate with the judges by sending message to the judges through the *Clarification System* in **PC^2**.
6. Since Visual Basic does not support console mode applications, VB programs should print output to the file "**output.txt**" instead of the standard output.

Problem A

Parentheses Balance

You are given a string consisting of parentheses ( ) and [ ]. A string of this type is said to be *correct*:

(a) if it is the empty string
(b) if A and B are correct, AB is correct,
(c) if A is correct, (A) and [A] is correct.

Write a program that takes a sequence of strings of this type and check their correctness. Your program can assume that the maximum string length is 128.

Input File

The file contains a positive integer n and a sequence of n strings of parentheses ( ) and [ ], one string a line.

Output

A sequence of Yes or No on the screen (standard output)

Sample Input

```
3
([])
(([()])))
([()[]()])()
```

Sample Output

```
Yes
No
Yes
```

Problem B

Coin Change

Suppose there are 5 types of coins: 50-cent, 25-cent, 10-cent, 5-cent, and 1-cent. We want to make changes with these coins for a given amount of money. For example, if we have 11 cents, then we can make changes with one 10-cent coin and one 1-cent coin, two 5-cent coins and one 1-cent coin, one 5-cent coin and six 1-cent coins, or eleven 1-cent coins. So there are four ways of making changes for 11 cents with the above coins. Note that we count that there is one way of making change for zero cent. Write a program to find the total number of different ways of making changes for any amount of money in cents. Your program should be able to handle up to 200 cents.

Input File

The input file contains a number for the amount of money in cents.

Output

The output is the number of different ways of making changes with the above 5 types of coins.

Sample Input

11

Sample Output

4

Problem C

Convex Hull of the Polygon

Suppose that a polygon is represented by a set of integer coordinates, $\{(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n), (x_0, y_0)\}$. Please find the convex hull of the polygon, where a convex hull is the minimum bounding convex polygon and "convex" means the angle between two consecutive edges is less than 180^o.

Input file

The input file contains a sequence of integer coordinates. The *i*th pair of numbers means the coordinates (x_i, y_i).

Output

The output should contain a sequence of integer coordinates, in which each coordinate is represented by a pair of integer numbers in a line.

Sample Input

0, 0
2, 0
1, 1
2, 2
0, 2
0, 0

Sample Output

0, 0
2, 0
2, 2
0, 2
0, 0

Problem D

Horse Step Maze

Maze search has been developed for a long time, and the related competition is held often. In this problem you have to design a program that simulates a computer mouse to find a path in a maze from a starting point to an ending point. To find such a path, the computer mouse must follow the knight steps as shown in Fig. 4.1. The numbers 1, 2, 3 and 4 indicate the sequence of four directions the computer mouse has to try while searching for a path. Specifically, the computer mouse must always try direction 1 first until it cannot continue searching the path to the ending point in the maze. When that happens, the computer mouse can backtrack and try direction 2. Similarly, if direction 2 can not work, direction 3 is tried and then direction 4.

The size of given maze is 9 by 9, and its coordinates are shown in Fig. 4.2. Initially, you are given the starting and ending points of a maze. Your program must print the walking path of the mouse if the ending point of the maze can be reached. Otherwise, “fail” should be printed.

Input File

The coordinate of the starting point followed by the coordinate of the ending point in a maze.

Output

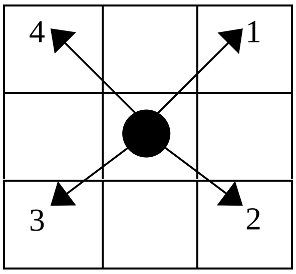
The coordinates of the walking path from the starting point to the ending point or “fail”.

Sample Input

(1,1)
(9,9)

Sample Output

(1,1), (2,2), (1,3), (2,4), (1,5), (2,6), (1,7), (2,8), (1,9), (2,8), (3,9), (4,8), (5, 9), (6,8), (7,9), (8,8), (9,9)



represents the location of computer mouse at present

represents the step that computer mouse can walk

The numbers 1,2,3 and 4 indicate the sequence of four directions the computer mouse must follow to find a path in the maze.

Fig. 4.1 The rule of the horse step

(1,1)								(1,9)
(9,1)								(9,9)

Fig. 4.2 The coordinates of the maze

Problem E

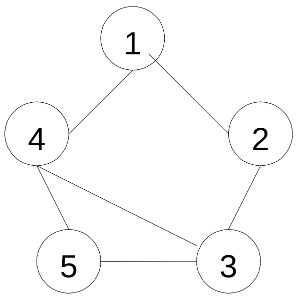
All Distinct Walks of length *n* among the first node and other nodes

A computer network can be represented as a graph. Let G = (V, E) be an undirected graph, V = (v_1, v_2, v_3, …v_m) represents all nodes, where *m* is the number of nodes, and E represents all edges. The first node is v_1 and the last node is v_m. The number of edges is k. Define the adjacency matrix A $(a_{ij})_{mxm}$ where

$$a_{ij} = 1 \quad if \ \{v_i, v_j\} \in E, \ otherwise \quad a_{ij} = 0$$

An example of the adjacency matrix and its corresponding graph are as follows:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$



Calculate

$$A^n = \underbrace{A \cdot A \cdots A}_{n}$$

and use the Boolean operations where 0+0 = 0, 0+1=1+0=1, 1+1=1, and 0•0=0•1=1•0=0, 1•1=1. The entry in row i and column j of A^n is 1 if and only if at least there exists a walk of length *n* between the i*th* and j*th* nodes of V. In other words, the distinct walks of length *n* between the i*th* and j*th* nodes of V may be more than one. Note that the node in the paths can be repetitive.

The following example shows the walks of length 2.

$$A^2 = A \cdot A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Write programs to do above calculation and print out all distinct walks of length *n*. (In this problem we let the maximum walks of length n be 5 and the maximum number of nodes be 10.)

Input File

The input file contains a number of test examples, the test examples are separated by –9999. Each test example consists of the number of nodes and the length of walks in the first row, and then the adjacency matrix.

Output

The output can be displayed directly on the screen to indicate all distinct walks of the length *n* from the first node.

Sample Input

```
5    2
0  1  0  1  0
1  0  1  0  0
0  1  0  1  1
1  0  1  0  1
0  0  1  1  0
-9999
5   3
0  1  0  1  0
1  0  1  0  0
0  1  0  1  1
1  0  1  0  1
0  0  1  1  0
```

Sample Output
Test example 1
(1,2,3)
(1,4,5)
(1,4,3)
Test example 2
(1,2,3,4)
(1,2,3,5)
(1,4,3,5)
(1,4,3,2)
(1,4,5,3)

Problem F

Schedule of Taiwan Baseball League

Taiwan baseball league and the sport division of the ministry of education of Taiwan are planning to organize a major event with world baseball championship. After several rounds of discussions among the members of the organizing committee, the details of the game plan are as follows.

1. There will be N teams invited to the competition, and N will be a power of 2. (N will also be less than or equal to 128.)
2. A round-robin competition and game schedule will be used. It means every team will play every other team exactly once.
3. Two cities will collectively host the games. Kaohsiung and Taipei will be the two cities to host the game, and Taipei Baseball Stadium and Kaohsiung Baseball Stadium (Lee-Der Stadium) will be used as the game fields.

Please write a program to produce a schedule for the games. The schedule needs to satisfy the following criteria.

a) The game schedule is round-robin. Suppose there are N teams to participate in the competition, you need to schedule games to be played in exactly N-1 days. In each day of the competition, every team will have to play exactly one game.
b) The games have to be scheduled evenly between Taipei and Kaohsiung. There will be equal number of games to be played each day between Taipei and Kaohsiung in your produced schedule.
c) The organizing committee also hopes to reduce the total number of travellings for the teams. Therefore, your schedule has to be the one with the minimum travellings for teams to travel between Taipei and Kaohsiung. For example, if a team plays in Taipei in Day 1, and then the same team plays in Kaohsiung in Day 2, we say that the travelling total is one. Your goal is to produce a schedule so that the total travelling combined with all the teams is the minimum. The fairness with the amount of travellings for each team is not required

by the organizing committee. We assume no travelling accounts in the first day of the game. The travelling total will be counted from day 2 of the games.

In the testing data, there will be K sets of data.

Input File

K
N_1
N_2
. . .
N_k

K is the number of testing data. N_i, i=1,..,K, are the total number of teams invited for the competitions. Totoally, there are K sets of data to be tested.

Output

T_1
T_2
. . .
T_k

For each testing case N_i, you need to report the total number T_i of travellings for the all the teams. The amount of travellings has to be minimum to be correct. The order in the report has to follow the input order with each testing case.

Sample Input

1
4

Sample Output

4

Explanations:
In the sample input, "1" means there is one testing case, and the next input means that there are totally 4 teams in the first testing case. The output "4" means 4 travellings needed for the given testing case.