The 10th provincial
programming contest
Birjand university

2024 ACM ICPC BIRJAND PROVINCIAL CONTEST

2024 ACM ICPC BIRJAND PROVINCIAL CONTEST

Birjand, 17 May 2024

Judges and Problem Setters

- Ali Jarkani
- Rohallah Hedayati
- Edris Hosseini Gol
- Sajjad Ahmadi
- Matin Hossein pour
- Bahram Abdorrahimi

2024 ACM ICPC BIRJAND PROVINCIAL CONTEST

Problem A - Grading System

Time limit: 1 second

Professor Hedayati has set specific rules for grading in his Operating Systems course based on student absences. The rules are as follows:

- If a student has zero absences, they will receive a grade of 20.
- If a student has exactly seven absences, they will retain their current grade X.
- Otherwise, for each day a student is absent (from 1 to 6 days or more than 7 days), their grade will decrease by exactly one point per day (if the grade drops below zero, it will be considered as zero).

Given the current grade X of a student and the number of absences N, determine the student's final grade.

Input

- The first line contains an integer X ($0 \leqslant X \leqslant 20$) which represents the student's current grade in the Operating Systems course.
- The second line contains an integer N ($0 \leqslant N \leqslant 100$) which represents the number of days the student was absent.

Output Print the final grade the student will receive after considering the number of absences.

Standard Input	Standard Output
14 0	20
6 7	6

Problem B - The Idiot of the Year!

Time limit: 1 second

There is just one basic rule in the Idiot of the Year Contest (IYC)! The contestant picks a random digit between 0 and 9, computes the factorial of the day of the year he/she is born, and counts how many times the digit picked appears in the factorial. The contestant with the highest count is the Idiot of the Year! For example, if you are born on the 5th of Mordad, which is the 129th day of the year, and you pick the digit 6, your score will be the number of times the digit 6 appears in 129! (that is $1 \times 2 \times 3 \times \ldots \times 129$).

The chief judge of IYC wants you to write a program to get an integer which is the day of the year a contestant is born on and a digit and report the number of times the digit appears in the factorial of the first number.

Input

The first line of the input contains a single integer T which is the number of test cases, followed by T lines each containing the data for a test case having two numbers. The first number is the day of the year a contestant is born, and the second one is the digit he/she has picked.

Output

The output contains T lines, each having one integer which is the number of times the digit appears in the factorial of the first number.

Standard Input	Standard Output
2 5 2 7 0	1 2

2024 ACM ICPC BIRJAND PROVINCIAL CONTEST

Problem C - Contest Preparation Schedule

Time limit: 1 second

Birjand University is organizing the 10th ICPC Provincial Competition, and the preparation schedule for this event is critical to its success. The organizing committee needs to create a schedule for multiple preparation activities, ensuring that all necessary tasks are completed on time without any overlap.

Each preparation activity has a specific start time and end time. The goal is to determine if it is possible to schedule all activities such that no two activities overlap.

Input

The input consists of:

- An integer n $(1 \leqslant n \leqslant 100)$ representing the number of preparation activities.
- n lines, each containing two integers $start$ and end $(0 \leqslant start < end \leqslant 1000)$ representing the start time and end time of each activity.

Output

Output "YES" if it is possible to schedule all activities without any overlap. Otherwise, output "NO".

Example

Input:

```
3
1 5
5 10
3 8
```

Output:

```
NO
```

Problem D - Stars

Time limit: 1 second

People generally don't care to give attention to stars in a moonlit night. In most cases the attention goes towards the moon. Sadly, you have to write a program now that can count the stars in the sky. For this problem a sky is a two-dimensional grid. Empty pixel is denoted by a '.' (ASCII value 46) and a non-empty pixel is denoted by a '*' (ASCII value 42). As a star is a very small object so it cannot occupy more than one pixel and in our sky two stars are never adjacent. So two or more adjacent non-empty pixels can denote some larger objects like moon, comet, sun or UFOs but they never represent a star. All the eight possible pixels around a pixel are adjacent to it. In the figure below the black pixel at the center has eight adjacent pixels. Of them three pixels are non-empty.

```
* . .
. * *
. . *
```

Input

The input file contains at most 1000 sets of inputs. The description of each set is given below:

Each set starts with two integer numbers r and c ($0 < r, c < 101$), which indicates the row and column number of the image to follow. Next r rows describe the sky as mentioned in the problem statement. Input is terminated by a line containing two zeroes.

Output

For each set of input produce one line of output. This line contains a decimal integer which denotes the number of stars in the given sky.

Standard Input	Standard Output
5 5 ....* ....* ...*. *.... 4 3 ... .*. ... *.* 0 0	1 3

Problem E - Parsing Real Numbers

Time limit: 1 second

Write a program that read a line of input and checks if the line contains a valid real number. Real numbers may have a decimal point, an exponent (starting with the character e or E), or both. Additionally, it has the usual collection of decimal digits. If there is a decimal point, there must be at least one digit on each side of the point. There may be a plus or minus sign in front of the number, or the exponent, or both (without any blank characters after the sign). Exponents are integers (not having decimal points). There may be blank characters before or after a number, but not inside it. Note that there is no bound on the range of the numbers in the input, but for the sake of simplicity, you may assume the input strings are not longer than 1000 characters.

Input

The first line of the input contains a single integer T which is the number of test cases, follow ed by T lines each containing the input line for a test case.

Output

The output contains T lines, each having a string which is LEGAL or ILLEGAL.

Standard Input	Standard Output
2 1.5e+2 3.	LEGAL ILLEGAL

Problem F - The Happy Worm

Time limit: 2 seconds

The Happy Worm lives in an $m \times n$ rectangular field. There are k stones placed in certain locations of the field. (Each square of the field is either empty, or contains a stone.) Whenever the worm sleeps, it lies either horizontally or vertically, and stretches so that its length increases as much as possible. The worm will not go in a square with a stone or out of the field. The happy worm cannot be shorter than 2 squares.

The question you are to answer is how many different positions this worm could be in while sleeping.

Input

The first line of the input file contains a single integer t ($1 \leqslant t \leqslant 11$), the number of test cases, followed by the input data for each test case. The first line of each test case contains three integers m, n, and k ($1 \leqslant m, n, k \leqslant 100000$). The input for this test case will be followed by k lines. Each line contains two integers which specify the row and column of a stone. No stone will be given twice.

Output

There should be one line per test case containing the number of positions the happy worm can be in.

Standard Input	Standard Output
1 5 5 6 1 5 2 3 2 4 4 2 4 3 5 1	9

Problem G - Baggage

Time limit: 2 seconds

An airline has two flights leaving at about the same time from ICPCity, one to city B and one to city A. The airline also has n counters where passengers check their baggage. At each counter there is a pair of identical baggage bins, one for city B and one for city A.

Just before the flights depart, each pair of baggage bins is moved by a motorized cart to a sorting area. The cart always moves two bins at a time, one for city B and one for city A. After all the bins have been moved, they line up in the sorting area like this:

B A B A B A ... B A

That is, there are 2n baggage bins in a row, starting with a bin for city B, then one for city A, and so forth. The task now is to reorder them so all the baggage bins for city A precede the baggage bins for city B. Then the bins can be loaded on the appropriate aircraft.

The reordering is done by moving pairs of adjacent baggage bins (not necessarily B then A), again via the motorized cart. For proper balance, the cart must always carry two bins, never just one. A pair of bins must always be moved to an empty space that is at least two bins wide. On the left of the first bin are some empty spaces that can be used as needed during the reordering.

When the reordering process begins, the bin locations are numbered from 1 (initially containing the leftmost B baggage bin) to $2n$ (initially containing the rightmost A baggage bin). There are $2n$ initially empty spaces to the left of the bins, numbered from 0 to $-2n+1$, as shown in Figure 1 for the case $n = 4$.

								B	A	B	A	B	A	B	A
-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8

Figure 1: Initial configuration of bins and empty spaces for $n = 4$

Given n, find a shortest sequence of moves that will reorder the bins so that all the A bins are to the left of all the B bins. At the end of the process, it is possible that the leftmost A bin is at some location other than 1, but the bins must be adjacent in a sequence of $2n$ locations.

Input

The input consists of a single test case, which consists of the integer n $(3 \leqslant n \leqslant 100)$.

Output

Display a shortest sequence of moves that will correctly reorder the bins. Each move is of the form "f to t", where f and t are integers representing the movement of the bins in locations f and $f+1$ to locations t and $t+1$. If multiple solutions are possible, display any one of them.

Standard Input	Standard Output
5	8 to -1 3 to 8 6 to 3 0 to 6 9 to 0
8	10 to -1 3 to 10 14 to 3 7 to 14 0 to 7 11 to 0 4 to 11 15 to 4

Problem H - Airport Construction

Time limit: 1 seconds

The tropical island nation of Piconesia is famous for its beautiful beaches, lush vegetation, cocoa and coffee plantations, and wonderful weather all year round. This paradise is being considered as a future location for the World Finals of the ACM International Collegiate Programming Contest (or at the very least a vacation spot for the executive council). There is only one small problem: the island is really hard to reach.

Currently, the fastest way to reach the island takes three days from the nearest airport, and uses a combination of fishing boat, oil tanker, kayak, and submarine. To make attending the ICPC World Finals slightly easier and to jump-start the island's tourism business, Piconesia is planning to build its first airport.

Since longer landing strips can accommodate larger airplanes, Piconesia has decided to build the longest possible landing strip on their island. Unfortunately, they have been unable to determine where this landing strip should be located. Maybe you can help?

For this problem we model the boundary of Piconesia as a polygon. Given this polygon, you need to compute the length of the longest landing strip (i.e., straight line segment) that can be built on the island. The landing strip must not intersect the sea, but it may touch or run along the boundary of the island. Figure A.1 shows an example corresponding to the first sample input.

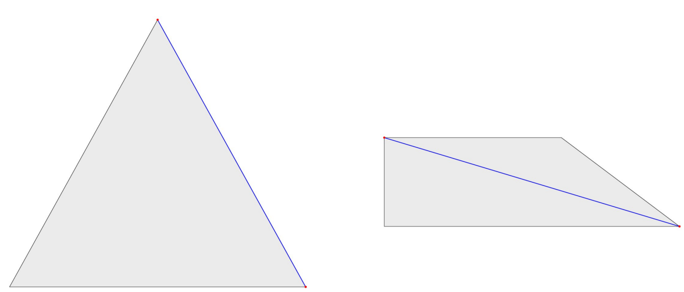

Figure 2: The island modeled as a polygon. The longest possible landing strip is shown as a thick line.

Input

The input starts with a line containing an integer n ($3 \leqslant n \leqslant 4$) specifying the number of vertices of the polygon. This is followed by n lines, each containing two integers x and y ($|x|, |y| \leqslant 10^6$) that give the coordinates (x, y) of the vertices of the polygon in counter-clockwise order. The polygon is simple, i.e., its vertices are distinct and no two edges of the polygon intersect or touch, except that consecutive edges touch at their common vertex. In addition, no two consecutive edges are collinear.

Output

Display the length of the longest straight line segment that fits inside the polygon, with an absolute or relative error of at most 10^{-6}

Standard Input	Standard Output
3 0 2017 -2017 -2017 2017 0	4510.149110617

2024 ACM ICPC BIRJAND PROVINCIAL CONTEST

Problem I - Final Price

Time limit: 1 second

Neverland has recently experienced a rapid rise in the inflation rate. The value of money is continuously decreasing, and citizens' purchasing power is lowered daily. The government is trying to control the inflation rate by testing various methods, such as reducing the amount of money in the economy by increasing interest rates and promoting investment, even in purchasing cars to be delivered in an unforeseeable future.

Despite these efforts, the inflation rate is still above 50%, and the prices are jumping up and down drastically every now and then. The Statistical Center of Neverland provides detailed information on the inflation rate and the average price change over time for a basket of goods commonly purchased by households. However, it is hard to calculate the actual price of a specific good at a given point in time using that information. The ICPC (International Center for Price Changes) is asking you to help the citizens of Neverland to calculate the prices for their desired goods after a specific period of time. The information provided to you is the initial price of a good at the start of a time period, and the daily price change for that good over the observed time period.

Input

The input consists of:

- An integer n $(1 \leqslant n \leqslant 1000)$ indicating the number of days the prices are observed.
- A positive integer indicating the initial price of the desired good on day 1.
- The next $n-1$ lines contain $n-1$ integers showing the daily change in the price of the good, in order from day 2 to day n. You can assume that the price of the good is always positive and never exceeds 1,000,000 in the observed period.

Output

In the output, print the final price of the good on day n.

Standard Input	Standard Output
4 110 -5 0 27	132
2 11 68	79

2024 ACM ICPC BIRJAND PROVINCIAL CONTEST

Problem J - To Add or to Multiply

Time limit: 4 second

The Industrial Computer Processor Company offers very fast, special purpose processing units tailored to customer needs. Processors of the *a*-*C*-*m* family (such as the 1-C-2 and the 5-C-3) have an instruction set with only two different operations:

A `add a`
M `multiply by m`

The processor receives an integer, executes a sequence of A and M operations (the program) that modifies the input, and outputs the result. For example, the 1-C-2 processor executing the program AAAM with the input 2 yields the output 10 (the computation is $2 \to 3 \to 4 \to 5 \to 10$), while the 5-C-3 processor yields 51 with the same program and input ($2 \to 7 \to 12 \to 17 \to 51$).

You are an *a*-*C*-*m* programmer assigned to a top secret project. This means that you have not been told the precise computation your program should perform. But you are given particular values p, q, r, and s and the following conditions:

1. The input is guaranteed to be a number between p and q.
2. The output must be some number between r and s.

Given an *a*-*C*-*m* processor and the numbers p, q, r, and s, your job is to construct the shortest *a*-*C*-*m* program which, for every input x such that $p \leqslant x \leqslant q$, yields some output y such that $r \leqslant y \leqslant s$. If there is more than one program of minimum length, choose the one that comes first lexicographically, treating each program as a string of As and Ms.

Input

The input contains several test cases. Each test case is given by a line with the six integers a, m, p, q, r, s as described above ($1 \leqslant a, m, p, q, r, s \leqslant 10^9$, $p \leqslant q$ and $r \leqslant s$).
The last test case is followed by a line with six zeros.

Output

For each test case, display its case number followed by the best program as described above. Display the word "`empty`" if the best program uses no operations. Display the word "`impossible`" if there is no program meeting the specifications.

Display the program as a sequence of space-separated strings, alternating between strings of the form `nA` and strings of the form `nM`, where $n > 0$. Strings of the former type indicate n consecutive A operations, and strings of the latter type indicate n consecutive M operations.

Follow the format of the sample output.

Standard Input	Standard Output
1 2 2 3 10 20	Case 1: 1A 2M
1 3 2 3 22 33	Case 2: 1M 2A 1M
3 2 2 3 4 5	Case 3: impossible
5 3 2 3 2 3	Case 4: empty
0 0 0 0 0 0	