AUST ACM Lab 02 Selection Contest 3 (Fall 21)

https://toph.co/c/7mbjr76

Toph

Schedule

The contest will run for **3h0m0s**.

Authors

The authors of this contest are edge555, hasibhossain, pz1971, rithyy, and Tash52.

Rules

This contest is formatted as per the official rules of ICPC Regional Programming Contests.

You can use Bash 5.0, Brainf*ck, C# Mono 6.0, C++11 GCC 7.4, C++14 GCC 8.3, C++17 GCC 9.2, C++20 GCC 12.1, C11 GCC 12.1, C11 GCC 9.2, Common Lisp SBCL 2.0, Erlang 22.3, Free Pascal 3.0, Go 1.18, Haskell 8.6, Java 1.8, Kotlin 1.1, Lua 5.4, Node.js 10.16, Perl 5.30, PHP 7.2, PyPy 7.1 (2.7), PyPy 7.1 (3.6), Python 2.7, Python 3.7, Ruby 2.6, Rust 1.57, Swift 5.3, and Whitespace in this contest.

Be fair, be honest. Plagiarism will result in disqualification. Judges' decisions will be final.

Notes

There are 7 challenges in this contest.

Please make sure this booklet contains all of the pages.

If you find any discrepencies between the printed copy and the problem statements in Toph Arena, please rely on the later.

Disclaimer

The contents of this contest have not been reviewed by Toph and do not necessarily represent Toph's views.

A. Dog and Bones

Alice has a robot dog, and she thinks her dog is smart. Bob doesn't think so. So bob gave Alice's dog a task,

1. There will be n bones around the dog (in positive xi and yi axis). And it should be able to pick the farthest bone from its (dog's) position (x,y).

2. There can be multiple bones in one coordinate. In this case, the bone that comes later (in input) will be placed on the last added bone in that coordinate. And robot dog will always choose the topmost bone.

For example – initially, there is no bone in position (2,3), then Bob adds a bone **B1** at this position. So now position (2,3) contains only one bone **B1**. If he adds another bone **B2,** in the same position, then **B2** will be placed on **B1**. So if (2,3) is the farthest point, then the robot dog will choose **B2** from this position.

Alice's dog solved the problem, but Bob was not happy with this only. Now Bob thinks if this dog can choose the **Kth farthest** point instead of the farthest point and picks its **topmost bone,** then he will say Alice's dog is the smartest.

Now Alice asks for your help. Write a program for Alice that will take the robot dog's position and positions of n bones and give the index (1 based) of the bone that the robot dog will pick as output. If there are different coordinates with the same distance, he has to choose the one which comes first in input. And if the **kth** farthest point doesn't exist, you have to print -1.

You can safely assume that no bone will be on the same coordinate as **the dog.**

Input

The first line of input will be **T**, the number of tests **(T<=50)** First line of every test case will contain two integers (x,y) **(0<=x,y<=10^9)**, the position of the robot dog. The next line will contain two positive integers **(n,k)**, where **(0<n,k<=100000)**. Then n line will contain two positive integers (xi,yi), coordinate of ith bone. **(0<=xi,yi<=10^9)**

Output

For each testcase your output format should be **"Case x: y"**. (without quotation). Where **x** defines number of testcase and **y** defines index of bone that will be picked by the robot dog if it exists, otherwise -1.

Samples

Input	Output
2 0 0 5 2 2 3 3 2 2 3 3 2 1 1 0 0 5 3 2 3 3 2 2 3 3 2 1 1	Case 1: 3 Case 2: -1

B. HAIL PEAKY BLINDERS

The Peaky Blinders is the most powerful gang in Birmingham. Almost every other gangs hate Peaky Blinders and want to destroy them.

Thomas Shelby, the leader of Peaky Blinders is aware of this. So, he has made a list of all other gangs which contains the name and power of each gang. He noticed that all gangs on his list have distinct power. Now Thomas wants to sort this list in descending order of power of the gangs.

Thomas is a busy guy so he has passed this list to you and asked you to sort it. Given the list, sort it as soon as you can.

Input

Input starts with T $(1 <= T <= 10)$, the number of test cases.

Each test case starts with N$(1 <= N <= 10^5)$denoting the number of gangs in following list.

The next N lines contain a string S denoting the gang name and integer X $(1 <= X <= 10^6)$denoting the power of the gang.

You can assume the name of each gang contains no more than 20 letters and doesn't contain any space.

Output

For each case print name of the gangs from your sorted list.

Samples

Input	Output
1 4 gang1 10 gang2 30 gang3 50 gang4 3	gang3 gang2 gang1 gang4

C. Valid Is Not Invalid

One day, Mr. Shoaib, a Delphi lover was writing a program that was producing an unexpected error repeatedly.

After failing to resolve the error, he sought help from his colleague, Mr. Tashfiq. After going through the code, Mr. Tashfiq found out that the error was “Invalid Parentheses Sequence”.

A sequence of parentheses is a string consisting of only opening and closing parentheses, i.e. ‘(‘ and ‘)’. A parentheses sequence is valid if it is in the form “(*)” where ***** can be either empty or another valid sequence of parentheses.

Given a sequence of opening and closing parentheses you will have to determine if it is a valid one.

Input

The input will contain a string of opening and closing parentheses. The string will be no longer than 25 characters.

Output

Print "Yes" if the input string contains a valid sequence of parentheses. Otherwise, print "No".

Samples

Input	Output
((((	No

Input	Output
((()()))	Yes

D. Kill the Enemy

You have n bullets and m enemies.

if you kill i-th enemy you will get a[i] points. And one bullet can kill only one enemy

Your task is to kill optimally to maximize the point.

Input

The first line of the input contains a single integer T denoting the number of test cases. The description of **T** (1 <= T <= 100) test cases follow. First line of each test case contains two integers **n** and **m** (1 <= n, m <= 1000)

Second line contains m space-separated integers **a[1], a[2], ..., a[m]** (1 <= a[i] <= 10^12)

Output

For each test case, print a single line containing one integer, the maximum point you can get.

Samples

Input	Output
3 3 3 1 2 3 2 3 17 16 23 2 5 1 1 1 1 1	6 40 2

E. Mango Game

Tashfiq is feeling the pressure for the upcoming ICPC reginal contest. To get some relief he plans to play a quick game with Akash bhai.

Tashfiq just loves mangoes and Akash bhai has brought an infinite amount of mangoes for his junior team mate.

Now they have set a basket and started throwing mangoes at it in turns with Tashfiq going first.

The probability of Tashfiq landing a mango in the basket each time is **A** and the probability for Akash bhai each time is **B**. Whoever lands the first mango in the basket wins.

You need to find the probability of Tashfiq winning the game.

Input

The first line will contain an integer **T** denoting the number of test cases. The next **T** lines will contain two real numbers each, **A** and **B**.

$1 <= T <= 1000$
$0 <= A < 1$
$0 <= B < 1$

Output

For each test, output a single line containing the probability of Tashfiq winning the game.
(see examples for clarification).

The answer will be considered correct if the absolute or relative error doesn't exceed 10^{-6}.
If the judges answer is **P** and contestant s answer is **Q** then $P - Q <= 10^{-6}$ should hold.

Samples

Input	Output
5 0 0.006 0.123 0.196 0.288 0.007 0 0 0.063 0	0 0.4171019 0.9829888 0 1

F. Design it

Let's design a new data structure!

In this new data structure, there will be a list and some operations. The operations are,

- **Insert Front X** — Insert **X** into the front of the list.
- **Insert Back X** — Insert **X** into the back of the list.
- **Erase Front** — Erase the first element of the list.
- **Erase Back** — Erase the last element of the list.
- **Get Front** — Print the first element of the list.
- **Get Back** — Print the last element of the list.
- **Get Max** — Print the maximum element of the list.
- **Get Min** — Print the minimum element of the list.

You need to design a data structure as described above and perform **Q** operations.

Input

The first line of the input will contain a single integer **Q**, denoting the number of operation.

The next **Q** lines will contain any of those operations described above.

$1 \leq Q \leq 10^5$

$1 \leq X \leq 10^6$

Assume that, every operation will be valid.

Output

Perform each operation as described in the problem statement.

Samples

Input	Output
8 Insert Back 8 Get Front Insert Front 9 Get Max Erase Back Insert Back 3 Get Back Get Min	8 9 3 3

G. Alam's Password

Alam wants to extract some passwords from a random string. A password can have any number of characters but it must contain at least one lowercase letter, one uppercase letter and a digit. He is looking at the string from left to right, and whenever he thinks he got a new password, he stores it in his database and starts checking from next character.

Given the string, can you determine how many passwords Alam will be able to extract from it?

Input

Each line will contain a string consisting only english alphabets and digits. The length of the string will be between 1 and 100. Read input till EOF.

Output

For each string print number of passwords Alam can extract from it.

Samples

Input	Output
hello1World2X aBAb222abD32Gggp aasdafsdfDSF	2 3 0

In the first case, the extracted passwords are “hello1W” and “orld2X”

In the second case, the extracted passwords are “aBAb2”, “22abD”,”32Gg”

In the third case there are no digits, so no password can be extracted.