2013-2014 ACM International Collegiate Programming Contest

Sponsored by talPor, Lourtec, Akdemia & 4-Geeks

USB Programming Contest Sartenejas

Universidad Simón Bolívar, Venezuela

November 02, 2013

Departamento de Computación y Tecnología de la Información
Universidad Simón Bolívar

This problem set should contain ten (10) problems on seventeen (17) non numbered pages. Please inform a runner immediately if something is missing from your problem set.
Welcome to the Programming Contest and enjoy!

Rules:

1. There are ten (10) problems for each team to be completed in five (5) hours.
2. All problems require you to read test data from the standard input and write results to the standard output.
3. You can use any of the following programming languages: C, C++ or Java. You can solve some problems in a language and others in another language. If you have already submitted a problem and it was rejected, you can submit it again in the same language or in another one.
4. The extensions to the sources files should be: **.c** for programs written in C, **.cpp** or **.cc** for those written in C++ and **.java** for those written in Java. Programs with a **.c** extension won't be compiled with a C++ compiler.
5. You can use any of the standard library functions that your chosen programming language provides. In addition, you can use the math library in C or C++. You can not use any other library that requires an extra flag to be passed to the compiler. If you do this, the judges will probably find a code "code compilation error" in your program.
6. Your program is not permitted to invoke any external program. For example, you may not use the system call ("grep xyz abc") to determine whether the file abc contains an occurrence of the string xyz. *Violation of this rule may lead to disqualification from the contest*
7. Output must correspond exactly to the provided sample output format, including (mis)spelling and spacing. Multiple spaces will not be used in any of the judges' output, except where explicitly stated.
8. Your solution to a problem should be submitted for judging using the Omicron system only. Once you have submitted the solution, it will reach the judges. The time it takes for your problem to be judged will depend on how busy the judges are. Once your submission has been judged, you will receive a message though Omicron indicating the judgment. the judgment maybe "Yes", meaning that your submission was judged to be correct. Otherwise you will get a message indicating the problem with your program. For example, the message may be "Wrong Answer", "Compilation Error", "Runtime Error", "Time Limit Exceeded" or some other that the judges consider necessary.
9. Programming style is not considered in this contest. You are free to code in whatever style you prefer. Documentation is not required. The judges will only test whether the input-output behavior of your program is correct or not.
   However, if your program takes more than a specific per-problem amount of time to execute for some input, it will be judged incorrect and a time-limit exceeded response will be given.
10. Programs written in Java must be named "Main.java" or Omicron will not be able to compile it.
11. Contestants can use their books, papers, or any other printed documentation or source code. But no soft copy will be allowed. No team is allowed to cooperate with any other team, by any means whatsoever, for solving the contest problems. *Any such attempt, if detected, will lead to immediate disqualification of all the teams involved.*
12. Judges' decisions are to be considered final.

Problem A
Another Enigma

Universidad Simón Bolívar

USB Local Programming Contest – Caracas November 02, 2013, Venezuela

Shortest problem statement ever:

You have a sequence V of size N and Q queries. Each query has a sequence W. For each query you have to say if W is a contiguous subsequence of V.

Input

You have T test cases. The first part of the input is the sequence V. The first line contains an integer N, and the next line contains N space-separated integers (V_i). The next line contains an integer Q, which is the number of queries for this test case. Q lines follow: each query has two lines of input, the first one corresponds to some integer S (the size of W), and the next line has S space separated integers (W_i).

- $0 < T < 10$
- $0 < N \leq 10^5$
- $0 < Vi \leq 10^4$
- $0 < Q \leq 10^3$
- $0 < S < 10^3$
- $0 < Wi < 10^4$

The input must be read from standard input.

Output

Output one single line with the number of the current test case (starting at 1) and then for each query of the test case, print a single line "YES" (without quotes) if the sequence of that query is a subsequence of V, otherwise print "NO" (without quotes.)

The output must be written to standard output.

Sample input	**Output for the sample input**
2 1 5 4 2 3 1 5 4 1 4 2 4 5 1 2 3 4 5 7 6 1 1 2 3 4 5 6	Case #1: YES NO NO Case #2: NO

Explanation of the first test case:
1 5 4 are the first three elements of the sequence.
1 4 is not a (contiguous) subsequence of V.
2 4 5 is not a contiguous subsequence of V (from left to right.)

Problem B
Buy Or Not

Universidad Simón Bolívar

USB Local Programming Contest – Caracas November 02, 2013, Venezuela

You are just about to buy a brand new television and want to take it inside your house so you can install it in your living room and relax watching Adventure Time. However you are unsure if you could pass the television through the corridors of your house all the way to your living room.

You are a very rich person and live in a big house with a lot of corridors. Each corridor has a certain width. You can assume that the length and height are big enough for any television to fit inside. Each corridor is connected to the next with a 90° angle (either positive or negative) and the corridors do not collapse into a circuit.

Given the widths of consecutive corridors and the width of the television you are about to buy, you should decide if the television fits through the corridors or not. If it fits, you should buy it. If it doesn't fit, there is really no point buying it.

Input

The input will start with a number T, the number of taste cases ($1 \leq T \leq 100$).

Each test case will begin with a line containing a number C ($1 \leq C \leq 20$), the number of corridors to consider. The next line contains C numbers C_i ($1 \leq C_i \leq 100$), the width of each corridor. Finally, the next line contains a number W ($1 \leq W \leq 300$), the width of the television you are considering buying.

The input must be read from standard input.

Output

For each test case output "I'll Buy it!" if the television fits through the corridor and "Bah! Didn't even want it!" is it doesn't. You should use 1e-3 precision when making comparisons.

The output must be written to standard output.

Sample input	**Output for the sample input**
2 2 27 64 125 3 64 27 64 126	I'll buy it! Bah! Didn't even want it!

Problem C
Chocolates

Universidad Simón Bolívar

USB Local Programming Contest – Caracas November 02, 2013, Venezuela

Mr. Yover Galarga is the owner of one of the biggest chocolate factories in the world. One day he was thinking about life and stuff and came up with the following problem and he is asking for your help.

His factory sells chocolate boxes of size $1 \times N$. Also, he has M different types of chocolate and infinite amounts of each type. The size of the i-th type of chocolate is $1 \times L_i$. Now, Mr. Yover is wondering in how many ways he can fill the box of chocolates. One way is different than the other if the number of chocolates used in the process, is different; the number of chocolates of some type is different or if the position of the chocolates are different. For more information, please go to the test cases explanation.

Input

You have to solve T test cases, where each test case consists of the following: the first line contains 2 number, N (the size of the chocolate box) and M (the number of different types of chocolate). After that, one line follows with M space-separated integers $L_1, L_2, \cdots, L_M$ (the sizes of the types of chocolates).

- The maximum size of the chocolate box (N) is 10^9.
- They have $0 < M \leq 20$ different types of chocolates.
- The size of each type of chocolate is $0 < L_i \leq 20$.

The input must be read from standard input.

Output

Output one single line with the number of the current test case (starting at 1) and the ways to fill the box of chocolates with the M types of chocolates (see sample output). The answer may be very big, so we are interested in the result modulo $10^9 + 7$.

The output must be written to standard output.

Sample input	**Output for the sample input**
4 4 3 1 2 3 10 2 2 8 1000000 1 3 10000 2 1 2	Case #1: 7 Case #2: 3 Case #3: 0 Case #4: 24223428

Explanation of the first test case:

```
| 1 | 1 | 1 | 1 |
| 1 | 1 | 2 |
| 1 | 2 | 1 |
| 2 | 1 | 1 |
| 2 | 2 |
| 1 | 3 |
| 3 | 1 |
```

Problem D
Drunken Buddies

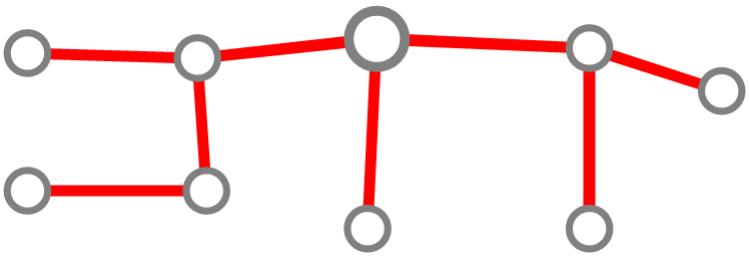
Universidad Simón Bolívar

USB Local Programming Contest – Caracas November 02, 2013, Venezuela

Scrappy and Coco are best friends. Recently, Coco has received a new hair cut and they have decided to drink until the sun goes up!.

After the tremendous party, Scrappy and Coco are so drunk that they start to travel in the subway system (which have a form of a tree) in random order, but after a while they are back in central station. They traveled in the following way: First, start at the central station. Second, pick a subway line at random and jump aboard the subway train. Third, in the station, they pick one of the subway lines they have not yet travelled on. If there is none left to explore at their current station, they take the subway line back on which they first came to the current station, until they eventually have travelled along all of the lines twice, once for each direction. Fourth, they are back at the central station.

Scrappy and Coco decided to travel in separated ways and record the "goes" and the "backs" in the following way: `GGBGGBBBGBGGBGBB`. `G` means that they travel away from the central station and `B` means that they travel back to the central station. This travel was done by Scrappy and represents the following diagram:



But Coco records this: `GBGGBGBBGGBGGBBB` this sequence represents the same trip, just in another order.

Scrappy and Coco ask for your help to write a program that given two travel sequences determine if the sequence can represent the same subway system.

Input

On the first line of input is a single positive integer N, telling the number of test scenarios to follow.

Each test scenario consists of two lines, each containing a string of the characters 'G' and 'B' of length at most 3000, both describing a correct exploration tour of a subway tree system.

The input must be read from standard input.

Output

The output should contain one line, that shows the text "same drunken trip." if the two strings may represent the same trip in the subway system, or the text "different drunken trip." if the trips cannot be the same.

The output must be written to standard output.

Sample input	**Output for the sample input**
2 GGBGGBBBGBGGBGBB GBGGGBBGBBGGBGBB GBGGBGBBGGBGGBBB GGBBGGGBBBGBGBGB	same drunken trip. different drunken trip.

Problem E
Enjoy the Skillrex and Shut Up!

Universidad Simón Bolívar

USB Local Programming Contest – Caracas November 02, 2013, Venezuela

Skillrex has noticed somebody has deciphered the complexity of his masterpieces. That's too mainstream, and being mainstream is not something you could easily live with.

Skillrex has come with a new plan: a new algorithm for creating his masterpieces. This is how it goes:

1. He choose a set of N notes.
2. He choose a number K of possible repetitions of each note.
3. When he uses a note n_i, next note should be different.
4. He builds a beat by choosing a sequence of K*N notes following the above rules.

A valid beat with 4 notes and 2 repetitions would be:

14134232

But this sequence would not:

11234234

Given N and K, can you determine how many possible beats does Skillrex have?

Input

The input will start with a number T, the number of taste cases ($1 \leq T \leq 100$).

Each test case will contain a line containing two numbers N ($1 \leq N \leq 10$) and K ($1 \leq K \leq 3$).

The input must be read from standard input.

Output

For each test case output a line with the number of possible beats Skillrex could masterfully create modulo 1000007.

The output must be written to standard output.

Sample input	**Output for the sample input**
7 3 2 2 2 5 3 10 3 5 2 6 2 2 1	30 2 606187 854583 39480 631586 2

Problem F
Foxy Business

Universidad Simón Bolívar

USB Local Programming Contest – Caracas November 02, 2013, Venezuela

Dog goes woof,
Cat goes meow,
Bird goes tweet
and mouse goes squeek.

Cow goes moo,
Frog goes croak
and the elephant goes toot.

Ducks say quack,
and fish go blub
and the seal goes ow ow ow.

But there's one sound
that no one knows.
What does the fox say?

Input

The input will contain several test cases.

Each test case will contain several lines of the form: "The X goes Y" (without quotes), where X and Y are non-empty strings of lower-case letters, possibly with spaces.

Each test case will end with the a line that has only "What does the fox say?" (without quotes).

The input must be read from standard input.

Output

For each test case output Y if a line a of the form "The fox goes Y" (without quotes) is part of the case's input. If no such line exists, simply output "will we ever know?" (without quotes).

The output must be written to standard output.

Sample input	**Output for the sample input**

`The cat goes meow`
`The dog goes woof woof`
`The fox goes ring a ding ding`
`The cow goes moo`
`What does the fox say?`
`The little duck goes quack`
`The big elephant goes toot`
`What does the fox say?`
	`ring a ding ding`
	`will we ever know?`

Problem G
Great Photo Day

Universidad Simón Bolívar

USB Local Programming Contest – Caracas November 02, 2013, Venezuela

It's the most expected day at your former High School! The great photo day!.

In this memorable day a group picture of all the students belonging to the same class is taken and published in the school's yearbook. For this end the kids from each class are organized in a single row sorted in ascending order by height, so afterwards the photographer can take care of immortalizing the moment.

In ancient times there where very few classes in the school, so a picture was taken for each class and all of them where published in the yearbook. However, since the amount of classes has exponentially grown due to city overpopulation, the school's director has decided to reduce the amount of pictures to be taken by overlaying rows: two rows of students can be overlaid if when sorted and placed together none of the students in the front row completely blocks any of the students in the rear row. Mathematically, if the heights of the students in row a and b, when sorted in ascending order, are $\{a_1, a_2, \ldots, a_m\}$ and $\{b_1, b_2, \ldots, b_m\}$, then row a can be overlaid by row b if $a_k > b_k$ for all k from 1 to m.

Since someone told your former School's director that you are a very clever computer scientist that aims to attend the ACM-ICPC world finals, he decided to hire you for building a program that given the specifications of all students in all the School's classes, computes the minimum amount of pictures that can be taken for including all classes while allowing overlaying.

Input

The input will consist of several test cases. The first line of each test case will consist of two numbers, n, and m ($2 \leq n, m \leq 100$); the number of classes in the school and the number of students per class. Each of the following n lines will contain m numbers separated by spaces, each describing the height h of one of the students in the class ($0 \leq h \leq 1000$). The final test case will have $n = m = 0$ and should not be processed.

The input must be read from standard input.

Output

For each test case print a single line: the minimum amount of pictures that can be taken for including all classes while allowing overlaying.

The output must be written to standard output.

Sample input	Output for the sample input
3 4 1 2 3 4 2 3 4 6 7 5 4 3 3 3 5 5 5 4 4 6 4 4 5 5 2 1 1 2 2 4 5 4 4 4 1 0 0	1 3 2

Problem H
Hoare Madness!

Universidad Simón Bolívar

USB Local Programming Contest – Caracas November 02, 2013, Venezuela

You are asked to implement an automatic verifier of the correctness of a given program. A triple Precondition Program Postcondition, also known as a Hoare triple, is considered correct if beginning at any state that satisfies the precondition and executing the program, the postcondition is ultimately satisfied. For example, the Hoare triple `{x >= 5} x := x * 2 {x >= 10}` is correct for all possible values of x, however the Hoare triple `{x >= 5} x := x * 2 {x > 10}` is not correct, for when x is 5, then after executing the program x would be 10, which is not greater than 10.

You will be given a program with the following concrete syntax:

```
Program := { Condition } StatementList { Condition }

Condition := Condition , Condition      // Conjunction
           | Expression < Expression    // Less than.
           | Expression > Expression    // Greater than.
           | Expression <= Expression   // Less or equal than.
           | Expression >= Expression   // Greater or equal than
           | Expression = Expression    // Equal
           | Expression != Expression   // Not equal

StatmentList := Statement ; StatementList
              | Statement

Statement := Variable := Expresion   // Assignment.

Expression := Value + Value   // Adition.
            | Value - Value   // Substraction.
            | Value * Value   // Multiplication.
            | Value / Value   // Integer Division.
            | Value

Value := Variable
       | Number

Variable := x
          | y
          | z
```

Note that the conditions do not have disjunctions or negations. However they do have conjunctions, meaning that both sub-conditions must be satisfied. Also note that every expression is forcibly a binary operation of values, or a single value. These values can be either literal numbers or variables. Only number from 0 to 100 will be valid for literals, and also as the possible range of values for variables. Finally note that the program can only have three variables: x, y and z.

You can assume that there will be at most 20 statements in the statement list. If at any point the evaluation of an expression results in an invalid value, you can assume the program was incorrect. If the evaluation of any part of the postcondition results in an invalid value, you can also assume the program incorrect.

Input

The first line of input will contain a single integer T, $1 \leq T \leq 10$, the number of test cases. T test cases follow. Each test case will be a program (Hoare triple) following the syntax described above. Every test case will be preceded by an empty line.

The input must be read from standard input.

Output

The output should consist of T lines, one for each test case. For each of the test cases you should output exactly on line containing "CORRECT" if the program is correct, or "INCORRECT" of the program was incorrect (quotes for clarity).

The output must be written to standard output.

Sample input	**Output for the sample input**
3 { x <= 5 } x := x * 2 { x <= 10 } { x <= 5 } x := x * 2 { x < 10 } { x <= y + z , y <= 10 , z <= 20 } y := y * 2 ; z := z * 2 ; { 2 * x <= y + z}	CORRECT INCORRECT CORRECT

Problem I
Inline Elephant Toast

Universidad Simón Bolívar

USB Local Programming Contest – Caracas November 02, 2013, Venezuela

While on a safari trip in an African savannah you find N gray elephants looking a bit confused. Strangely, each elephant is holding a glass of champagne with its trunk. As soon as they see you approach, one of them shouts to you and asks for help.

As it turns out, this group of big, gray and smart mammals is having a celebration because all of them advanced to the world finals of an important international programming competition. They want to make a toast: they will raise their glasses, clink them together with the other elephants, and drink.

To make things a bit more interesting, however, they've decided to do this while standing in a straight line. First, the elephants will pair up in an arbitrary way and each pair of elephants will clink their glasses together (all at the same time). Then, they will choose another way of pairing up and clink their glasses again. They will continue doing this until they've exhausted all possible ways of pairing up.

But not all ways of pairing up are possible. If the trunks of the elephants cross paths in the air they might become entangled, so they need to avoid that. For example, suppose there are four elephants aligned in this way:

Dumbo Jumbo Catty Giddy

It's perfectly fine if Dumbo pairs up with Jumbo and Catty pairs up with Giddy at the same time. Dumbo could also pair up with Giddy while Jumbo pairs up with Catty. But if Dumbo paired up with Catty and Jumbo paired up with Giddy, their trunks would cross in the air, so that's not a possible way to pair up.

Despite being very good programmers, they have no idea how to count the number of possible ways to pair up. That's why they're confused and need you to solve the problem for them.

Note: this breed of elephants has very long trunks, so even the elephants at both ends of the line can reach each other if they decide to clink their glasses.

Input

The input will contain several test cases. Each test case consists of an even number $1 \leq N \leq 50$ in its own line.

A single 0 will follow the last test case.

The input must be read from standard input.

Output

For each test case output the number of ways the N elephants can pair up and clink their glasses together without getting their trunks entangled.

The output must be written to standard output.

Sample input	**Output for the sample input**
2 4 6 0	1 2 5

Problem J
Jabba The Hutt

Universidad Simón Bolívar

USB Local Programming Contest – Caracas November 02, 2013, Venezuela

For many years, Mos Eisly city has been the host of the mythical "Shoot Jabba The Hut" shooting tournament (this of course, many years after Jabba the Hut were killed by the hand of princess Leia), where big prizes are awarded to the very best in each category. Categories range from "Apprentice" to "Grand Master". Everyone pays special attention to the "Grand Master" category, for it invites the best shooters in the entire Galaxy.

This edition of "Shoot Jabba The Hut" has two very important competitors: Jaina Solo and Cade Skywalker, both descendants of captain Han Solo and Luke Skywalker respectively. Both competitors hold an intense rivalry, not only to prove who the best Jedi is, but to impose a superior linage.

The rules of the competition are as follows:

1. Each player has a single round, each consisting of several subrounds.
2. Each subround consists of D targets, each with value V_i.
3. The score for each target is the targets value, multiplied by the target multiplier.
4. At first, the multiplier is 1 and for each target hit, it increases by 1.
5. If a target is missed, the multiplier goes back to 1.

However, the Hutts still rule the planet and organized crime is under their control. One of the many businesses they manage is gambling and it could not be the case that this tournament was free of bets.

Watto, the old junk salesman is a fan of gambling and has contacts with the competition organizers. They have told him the arrangement of targets that will be presented to each competitor and for how long. Besides, Watto know quite precisely the amount of targets that Jaina and Cade can hit each second. Needless to say, two Jedi master such as them never miss a shot. Watto is in great debt with the Hutts and will lose his beloved wings to them if he does not make a smart bet. So, he wants to know the greatest score that Jaina and Cade can achieve in order to make his decision.

Input

The first line of input will be a positive integer Q ($Q \leq 100$), the number of test cases.

Each test case will begin with a line with four integers N, M, J, C ($1 \leq N, M \leq 100$; $1 \leq J, C, \leq 20$), where N is the amount of subrounds that Jaina will play, M the amount of subrounds Cade will play, J and C are the amount of targets that Jaina and Cade can hit per second, respectively.

The following $N + M$ lines will contain the information for Jaina's and Cade's subrounds. Each subround begins with two integers D and T ($1 \leq D \leq 2000$; $1 \leq T \leq 20$), where D is the amount of targets present in the subround and T is the time the subround lasts. Following are D integers, with the value of each target in that subround.

The input must be read from standard input.

Output

For each test case, output a line with the competitor that Watto would bet on. In case of a tie, you should output "Watto, run!" (without quotes).

The output must be written to standard output.

Sample input	**Output for the sample input**
2 2 3 2 1 5 4 5 2 7 6 1 3 2 1 2 3 3 1 1 2 3 5 2 6 2 7 8 7 2 4 5 4 4 4 10 10 4 4 1 2 3 4 3 3 5 6 7 2 2 8 9 1 1 10 4 4 1 2 3 4 3 3 5 6 7 2 2 8 9 1 1 10	Jaina Watto, run!