2013-2014 ACM International Collegiate Programming Contest

Sponsored by talPor, Lourtec, Akdemia & 4-Geeks

USB Programming Contest
Warm Up Session / Sartenejas

Universidad Simón Bolívar

Centro de Estudiantes de Ingeniería de Computación

Universidad Simón Bolívar, Venezuela

November 02, 2013

Departamento de Computación y Tecnología de la Información
Universidad Simón Bolívar

This problem set should contain three (3) problems on four (3) 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 three (3) problems for each team to be completed in one hour and a half (1.5).
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
Are You Kidding Me?

Universidad Simón Bolívar

USB Local Programming Contest – Warm Up Session, Caracas November 03, 2012, Venezuela

No fancy statement here. You are given two integers a and b whose absolute value is less than 1000. Output their sum.

Input

The input contains several testcases. Each is specified by integers a and b.

The input must be read from standard input.

Output

For each test case output the sum of the numbers a and b.

The output must be written to standard output.

Sample input	**Output for the sample input**
2 4 -3 3 -400 999	6 0 599

Problem B
Baby Hipster

Universidad Simón Bolívar

USB Local Programming Contest – Warm Up Session, Caracas November 03, 2012, Venezuela

A common intelligence test for babies consists of presenting them with a set of three dimensional figures, and a base with slots that match the figures. The goal is to insert each figure in the corresponding matching slot in the base.

However, one of the babies, refuses to place the three dimensional rectangle in it's corresponding slot. After all, that is way too mainstrean. Instead, the baby insists in placing the rectangle in the cylinder's slot.

Curious about this behaviour you decide to implement a program which tests if the the rectangle actually fits in the cylinder's slot.

Input

The input consists of many test cases. Every tast case starts with and integer number r, the radious of the base of the cylinder slot. Input is terminated by $r = 0$. Otherwise, $1 \leq r \leq 1000$. Then follow 2 integer numbers w and l specifying the width and the length of the base of the rectangle. You can assume that the baby will not rotate the rectangle, so it's height is useless.

The input must be read from standard input.

Output

Output for each test case whether the rectangle will fit in the cylinders slot or not. Adhere to the format shown in the sample output. A rectangle which touches the border of the slot, without intersecting it, is considered to fit in the slot.

The output must be written to standard output.

Sample Input

```
38 40 60
35 20 70
50 60 80
0
```

Sample Output

```
Rectangle 1 fits in the slot.
Rectangle 2 does not fit in the slot.
Rectangle 3 fits in the slot.
```

Problem C
Co-Magic Sets

Universidad Simón Bolívar

USB Local Programming Contest – Warm Up Session, Caracas November 03, 2012, Venezuela

We define a magic set as a sequence of integer numbers where the number of ocurrences of each element is equal to the element itself. For example: $\{1, 2, 3, 3, 2, 3\}$ is a magic set, because 1 occurs a single time, 2 occurs twice and 3 occurs thrice.

Furthermore, we define a co-magic set as a sequence of integer numbers where if an element s_i has c_i occurences, then an element woth value c_i must occur in the sequence, and it must have s_i ocurrences. For example: $\{1, 3, 1, 2, 2, 1\}$ is a co-magic set. Note that every magic set is also a co-magic set.

You are given a secuence of integer numbers. Your job is to find out if it is a magic-set, co-magic set or neither.

Input

The first line of input will contain a single integer T, $1 \leq T \leq 1000$, the number of test cases. T test cases follow. The first line of each test case will consist of a single integer N, $0 \leq N \leq 210$, the size of the sequence. The following line will contain N integers, separated by whitespaces, being the elements of the sequence. Each element is a positive integer number lower than or equal to 20.

The input must be read from standard input.

Output

The output should contain a single line for each test case, deciding whether the sequence is a magic set, a co-magic set or neither. Adhere to the format shown in the sample output.

The output must be written to standard output.

Sample Input

```
3
6
1 2 3 3 2 3
6
1 3 1 2 2 1
6
1 1 2 2 2 3
```

Sample Output

```
Magic!
Co-Magic!
Regular.
```