



**V MARATON INTERNA DE PROGRAMACION
UNIVERSIDAD KONRAD LORENZ**

25 de Abril de 2009

PROBLEMAS

Elaborado por: Hector Florez
Basado de www.acis.org.co, www.acm.org

Problem 1

7 segments

Source file name: segments.c, segments.cpp or segments.java

Input: segments.in

Output: standar output

A 7 segments display is a component built whit diodes LED (Light emisor diode) which allow showing decimal numbers.

Next figure shows a 7 segments display and then shows the representation of each digit.



Input

Input consists in several cases. Each one, contain a decimal number and other decimal number between 1 and 5 separated by a space.

Output

For each case, you must print the first number with the number of characters determined by the second number for each segment. In case where the segment is off, you must print a space.

Sample input	Sample Output
3 3 8 2	--- --- --- -- --- --

Problem 2

Electrical Outlets

Source file name: outlets.c, outlets.cpp or outlets.java

Input: outlets.in

Output: standar output

Roy has just moved into a new apartment. Well, actually the apartment itself is not very new, even dating back to the days before people had electricity in their houses. Because of this, Roy's apartment has only one single wall outlet, so Roy can only power one of his electrical appliances at a time.

Roy likes to watch TV as he works on his computer, and to listen to his HiFi system (on high volume) while he vacuums, so using just the single outlet is not an option. Actually, he wants to have all his appliances connected to a powered outlet, all the time. The answer, of course, is power strips, and Roy has some old ones that he used in his old apartment. However, that apartment had many more wall outlets, so he is not sure whether his power strips will provide him with enough outlets now.

Your task is to help Roy compute how many appliances he can provide with electricity, given a set of power strips. Note that without any power strips, Roy can power one single appliance through the wall outlet. Also, remember that a power strip has to be powered itself to be of any use.

Input

Input will start with a single integer $1 \leq N \leq 20$, indicating the number of test cases to follow. Then follow N lines, each describing a test case. Each test case starts with an integer $1 \leq K \leq 10$, indicating the number of power strips in the test case. Then follow, on the same line, K integers separated by single spaces, $O_1 O_2 \dots O_k$, where $2 \leq O_i \leq 10$, indicating the number of outlets in each power strip.

Output

Output one line per test case, with the maximum number of appliances that can be powered.

Sample input	Sample Output
3	7
3 2 3 4	31
10 4 4 4 4 4 4 4 4 4	37
4 10 10 10 10	

Problem 3

Base Comparator

Source file name: bcomp.c, bcomp.cpp or bcomp.java

Input: bcomp.in

Output: standar output

DigiCircuits Inc. is a software company that develops software simulators for digital circuits. A very frequently used component of its software, named the comparator, is a simulated circuit that compares numbers expressed in different numerical bases. More exactly, this component receives two numbers, each one in a possible different base, and decides if the first number is less than, equal to or greater than the second number. The numerical bases that may appear vary from 1 to 9. Remember that a number expressed in base b uses only digits less than b . Your task is to develop a program that simulates the function of the comparator component.

Input

The input file contains several test cases, each one of them in a separate line. Each test case has four numerical strings, each two of them separated by a blank character, say $s\ b\ t\ c$. Strings b and c are one-character strings. They represent the bases for the first and third strings s and t , respectively.

The end of the input is denoted by the end of the input file.

Output

Output text for each input case is presented in the same order that input is read. For each test case the answer must be a left aligned answer-character corresponding to the input. This character must be $<$, $=$ or $>$, accordingly to the fact that the first string represents a numerical value less than, equal to or greater that the represented by the third string.

Sample input	Sample Output
54 6 71 8	<
110 2 6 7	=
3 4 3 9	=
14 7 1000 2	>

Problem 4

Speed Limit

Source file name: speed.c, speed.cpp or speed.java

Input: speed.in

Output: standar output

Bill and Ted are taking a road trip. But the odometer in their car is broken, so they don't know how many miles they have driven. Fortunately, Bill has a working stopwatch, so they can record their speed and the total time they have driven. Unfortunately, their record keeping strategy is a little odd, so they need help computing the total distance driven. You are to write a program to do this computation.

For example, if their log shows

Speed in miles per hour	Total elapsed time in hours
20	2
30	6
10	7

this means they drove 2 hours at 20 miles per hour, then $6-2=4$ hours at 30 miles per hour, then $7-6=1$ hour at 10 miles per hour. The distance driven is then $(2)(20) + (4)(30) + (1)(10) = 40 + 120 + 10 = 170$ miles. Note that the total elapsed time is always since the beginning of the trip, not since the previous entry in their log.

Input

The input consists of one or more data sets. Each set starts with a line containing an integer n , $1 \leq n \leq 10$, followed by n pairs of values, one pair per line. The first value in a pair, s , is the speed in miles per hour and the second value, t , is the total elapsed time. Both s and t are integers, $1 \leq s \leq 90$ and $1 \leq t \leq 12$. The values for t are always in strictly increasing order. A value of -1 for n signals the end of the input.

Output

For each input set, print the distance driven, followed by a space, followed by the word "miles".

Sample input	Sample Output
3	170 miles
20 2	180 miles
30 6	90 miles
10 7	
2	
60 1	
30 5	
4	
15 1	
25 2	
30 3	
10 5	
-1	

Problem 5

Age

Nombre del archivo fuente: age.c, age.cpp or age.java

Entrada: age.in

Salida: Salida Estándar

Juan wants to calculate the age of the people he knows given a particular date. Based in two dates, one is the birth date and other one a date considered the current date, make a program to calculate the age in each case.

Input

The input contains several test cases. Each line contains one case. There are 2 dates at each line. The first date is the birth date and the next is the current date. The date format is yy-mm-dd. Those dates are separated by one space. The current date always will be greater than the birth date.

Salida

For each case, calculate the age

Sample input	Sample Output
1980-02-05 2008-05-06	28
1940-05-07 2000-03-04	59

Problem 6

Generalized Matrioshkas

Source file name: matriosh.c, matriosh.cpp or matriosh.java

Input: matriosh.in

Output: standar output

Vladimir worked for years making matrioshkas, those nesting dolls that certainly represent truly Russian craft. A matrioshka is a doll that may be opened in two halves, so that one finds another doll inside. Then this doll may be opened to find another one inside it. This can be repeated several times, till a final doll -that cannot be opened- is reached.

Recently, Vladimir realized that the idea of nesting dolls might be generalized to nesting toys. Indeed, he has designed toys that contain toys but in a more general sense. One of these toys may be opened in two halves and it may have more than one toy inside it. That is the new feature that Vladimir wants to introduce in his new line of toys.

Vladimir has developed a notation to describe how nesting toys should be constructed. A toy is represented with a positive integer, according to its size. More precisely: if when opening the toy represented by m we find the toys represented by n_1, n_2, \dots, n_r , it must be true that $n_1 + n_2 + \dots + n_r < m$. And if this is the case, we say that toy m contains directly the toys n_1, n_2, \dots, n_r . It should be clear that toys that may be contained in any of the toys n_1, n_2, \dots, n_r are not considered as directly contained in the toy m .

A generalized matrioshka is denoted with a non-empty sequence of non zero integers of the form: $a_1 a_2 \dots a_N$, such that toy k is represented in the sequence with two integers $-k$ and k , with the negative one occurring in the sequence first that the positive one.

For example, the sequence

$-9 - 7 - 2 2 - 3 - 2 - 1 1 2 3 7 9$

represents a generalized matrioshka conformed by six toys, namely, 1, 2 (twice), 3, 7 and 9. Note that toy 7 contains directly toys 2 and 3. Note that the first copy of toy 2 occurs left from the second one and that the second copy contains directly a toy 1. It would be wrong to understand that the first -2 and the last 2 should be paired.

On the other hand, the following sequences do not describe generalized matrioshkas:

$-9 - 7 - 2 2 - 3 - 1 - 2 2 1 3 7 9$

because toy 2 is bigger than toy 1 and cannot be allocated inside it.

$-9 - 7 - 2 2 - 3 - 2 - 1 1 2 3 7 - 2 2 9$

because 7 and 2 may not be allocated together inside 9.

$-9 - 7 - 2 2 - 3 - 1 - 2 3 2 1 7 9$

because there is a nesting problem within toy 3.

Your problem is to write a program to help Vladimir telling good designs from bad ones.

Input

The input file contains several test cases, each one of them in a separate line. Each test case is a sequence of non zero integers, each one with an absolute value less than 10^7 .

Output

Output texts for each input case are presented in the same order that input is read. For each test case the answer must be a line of the form

:-) Matrioshka!

if the design describes a generalized matrioshka. In other case, the answer should be of the form

:-(Try again.

Sample input	Sample Output
-9 -7 -2 2 -3 -2 -1 1 2 3 7 9	:-) Matrioshka!
-9 -7 -2 2 -3 -1 -2 2 1 3 7 9	:-(Try again.
-9 -7 -2 2 -3 -1 -2 3 2 1 7 9	:-(Try again.
-100 -50 -6 6 50 100	:-) Matrioshka!
-100 -50 -6 6 45 100	:-(Try again.
-10 -5 -2 2 5 -4 -3 3 4 10	:-) Matrioshka!
-9 -5 -2 2 5 -4 -3 3 4 9	:-(Try again.