

Factorials !!!

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

Input: factorials.in

Output: standar output

Definition

$n!!! = n(n-k)(n-2k) \dots (n \bmod k)$, if k doesn't divide n

$n!!! = n(n-k)(n-2k) \dots k$, if k divides n (There are k marks ! in the both cases).

For example, $10 \bmod 3 = 1$; $3! = 3 \times 2 \times 1$; $10!!! = 10 \times 7 \times 4 \times 1$;

Given numbers n and k we have calculated a value of the expression in the first definition. Can you do it as well?

Input

There will be multiple cases. Each test case will be contained on one line. Each line will start with the followed by exactly one space, then k exclamation marks.

Output

For each test case you should print one line of output which contains one number – $n !!!$ (there k marks ! here)

Sample input	Sample Output
3 !	6
10 !!!	280
9 !!	945

Odd or Even

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

Input: odd.in

Output: standar output

There are several versions of Odd or Even, a game played by competitors to decide random issues. In one of the versions, for two players, the game starts with each player calling either odds or evens. Then they count to three (some people chant “Once, twice, three, SHOOT!”). On three, both players hold out one of their hands, showing a number of fingers (from zero to five). If the fingers add to an even number, then the person who called evens wins. If the fingers add to an odd number, then the person who called odds wins.

John and Mary played several games of Odd or Even. In every game John chose odds (and, Mary chose evens). During the games each player wrote down, in small cards, how many fingers he/she showed, using one card for each game Mary used blue cards, John used red cards. Their objective was to be able to re-check the results later, looking at the cards for each game. However, at the end of the day John dropped the deck of cards, and although they could separate the cards by color, they are now out of order. Given the set of numbers written on red cards and on blue cards, you must write a program to determine the minimum number of games that Mary certainly won.

Input

The input contains several test cases. The first line of a test case contains an integer N representing the number of games played ($1 < N < 100$). The second line of a test case contains N integers X_i , indicating the number of fingers shown by Mary in each of the games ($0 \leq X_i \leq 5$, for $1 \leq i \leq N$). The third line of a test case contains N integers Y_i , indicating the number of fingers shown by John in each of the games ($0 \leq Y_i \leq 5$, for $1 \leq i \leq N$). The end of input is indicated by $N = 0$.

Output

For each test case your program must write one line, containing one integer, indicating the minimum number of games that Mary certainly won.

Sample input	Sample Output
3	0
1 0 4	3
3 1 2	
9	
0 2 2 4 2 1 2 0 4	
1 2 3 4 5 0 1 2 3	
0	

Uncle Jack

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

Input: uj.in

Output: standar output

Dear Uncle Jack is willing to give away some of his collectable CDs to his nephews. Among the titles you can find very rare albums of Hard Rock, Classical Music, Reggae and much more; each title is considered to be unique. Last week he was listening to one of his favorite songs, Nobody's fool, and realized that it would be prudent to be aware of the many ways he can give away the CDs among some of his nephews.

So far he has not made up his mind about the total amount of CDs and the number of nephews. Indeed, a given nephew may receive no CDs at all.

Please help dear Uncle Jack, given the total number of CDs and the number of nephews, to calculate the number of different ways to distribute the CDs among the nephews.

Input

The input consists of several test cases. Each test case is given in a single line of the input by, space separated, integers N ($1 \leq N \leq 10$) and D ($0 \leq D \leq 25$), corresponding to the number of nephews and the number of CDs respectively. The end of the test cases is indicated with $N = D = 0$.

Output

The output consists of several lines, one per test case, following the order given by the input. Each line has the number of all possible ways to distribute D CDs among N nephews. The output must be written to standard output.

Sample input	Sample Output
1 20	1
3 10	59049
0 0	

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	>