Greetings From Globussoft

- Given below are 5 Programming questions, you have to solve any 3 out of 5 questions.

- These 5 questions you can attempt in any technology like C/C++, java, .Net

- To solve these 5 questions you've max. 3 hours.

- While solving these questions you are not allowed to use any **Search Engine** like Google, Yahoo, Bing …

All the best for your test

Globussoft

Question: - 1

The Factory of Computer Enhaced Numbers (FCEN) has asked its Development Comitee (DC) to come up with a way to handle numbers written in base 17. As everybody knows, base 17 is very important for many scientific applications, as well as for engineering and other practical uses. Numbers in base 17 can be tough, but are kind and soft if treated appropiately.

Numbers in base 17 are written by using a set of 17 characters: digits 0 to 9 with their usual values, and uppercase letters A to G that have values from 10 to 16, respectively. Base 17, probably because its basement on a prime number, does not require numbers to start with a non-zero digit, so each number has many representations. For instance, the decimal number 117 can be written as 6F , but also as 06F or even 00000006F .

Because of this leading-zeroes thing, heptadecimal numbers are hard to compare. As a member of the FCEN-DC, you were asked to write a program that helps in this difficult and challenging task.

Input

The input contains several test cases. Each test case is described in a single line that contains two non-empty strings of at most 105 heptadecimal digits, separated by a single space. The last line of the input contains two asterisks separated by a single space and should not be processed as a test case.

Output

For each test case output a single line with the sign "<" if the first heptadecimal number is smaller than the second one, the sign ">" if the first heptadecimal number is greater than the second one, or the sign "=" if both heptadecimal numbers are equal.

Example:

Input

006F B3B
0000 0
* *

Output

<
=

Question: - 2

When dealing with a long task, computers often provide a progress indicator to help users estimate how much longer they will have to wait. This is especially useful when copying a large number of data files from one drive to another.

In the Institute of Computer Power Control (ICPC) are very concerned about their brand new file copier, which they think will change forever the way people copy files. While this is a great accomplishment for the engineers in ICPC, the lack of a progress indicator is threatening the future of the project and the well being of most computer users around the world! The Supremum Principal Director Manager of ICPC has called you personally to ensure you are up for the task. The interface provided by the developing team of the file copier only gives two integers M and N . M is the number of files that have already been copied, and N is the total number of files to be copied. Using this information, you must write a module that displays the progress indicator.

The indicator must be drawn as a string of exactly 20 characters. The first K of them must be asterisks ("*") and the rest must be hyphens ("-"). The number K must be chosen in such a way that K/20 correctly approximates M/N ; this means that the distance between the two mentioned fractions is minimum. If there is more than one possible value for K, the greatest one must be chosen.

Also, for more precision, a number P without leading zeroes and followed by a percentage sign ("%") must be written on top of the described indicator. Since the goal is to represent the finished percentage, the number P must be such that P/100 correctly approximates M/N , with the same policy as before. The finished percentage must be centered on top of the display. This means that if possible, the same number of display characters ("*" or "-") must be seen to the left and to the right of the percentage; if this is not possible, exactly one extra character must be seen to the left.

Input

The input contains several test cases. Each test case is described in a single line that contains two integers M and N as explained above ($0 \le M \le N \le 109$ and $N \ne 0$). These values are separated by a single space. The last line of the input contains the number -1 twice separated by a single space and should not be processed as a test case.

Output

For each test case output a single line with exactly 20 characters representing the mentioned display.

Example:

Input

2 5
2 6
0 10
-1 -1

Output

```
********-40%--------
*******--33%--------
---------0%---------
```

Question: - 3

A palindrome is a symmetrical string, that is, a string read identically from left to right as well as from right to left. You are to write a program which, given a string, determines the minimal number of characters to be inserted into the string in order to obtain a palindrome. As an example, by inserting 2 characters, the string "Ab3bd" can be transformed into a palindrome ("dAb3bAd" or "Adb3bdA"). However, inserting fewer than 2 characters does not produce a palindrome.

Input

The first line contains one integer: the length of the input string N, 3≤N≤5000. The second line contains one string with length N. The string is formed from uppercase letters from 'A' to 'Z', lowercase letters from 'a' to 'z' and digits from '0' to '9'. Uppercase and lowercase letters are to be considered distinct.

Output

The first line contains one integer, which is the desired minimal number.

Example:

Input

5
Ab3bd

Output
2

Question: - 4

Envido is an important part of the typical argentinian game Truco. It is even more important when Falta Envido is called, because in this case the winner of the Envido wins the entire game. In this problem you are asked to calculate the envido advantage of a cheater. Truco is played with a limited deck of spanish cards. This deck contains four suits: espada, basto, copa and oro. There are 10 cards of each suit: 1, 2, 3, 4, 5, 6, 7, 10, 11 and 12. Cards 10, 11 and 12 have an envido value of 0, while the envido value for any of the other cards is simply its number. Each player holds three cards. The winner of an Envido is the player with the higher envido score. When calculating his envido score each player can either: 1. select a single card and have as envido score the envido value of the card; or 2. select two cards of the same suit and have as envido score 20 plus the sum of the envido value of each selected card.

Players always select the card or cards that yield the highest envido score. For instance, a player having a 10 and a 2 of the same suit, together with a 5 of a different suit, would inform an envido score of 22 because 20 + 0 + 2 = 22.

A cheater playing Truco thinks that he can change one of his three cards without anybody else in the table noticing. If he changes more than one card, he surely will be discovered, so he does not do that. Since even changing one card is risky, he only does it when the envido score of the resulting hand is much higher than the one of his original hand. Given the cards the cheater has in hand, you must calculate the maximum increase he can get in the envido score by exchanging one of his cards with a card from the deck. Notice that the resulting hand must be formed by two of his original cards and one new card selected from the deck, and that he cannot have two equal cards (same number and suit).

Input

The input contains several test cases. Each test case is described in a single line that contains six values N_1 , S_1 , N_2 , S_2 , N_3 , and S_3 separated by single spaces. Each pair (N_i , S_i) describes

a card in the hand of the cheater (1 ≤ i ≤ 3), where Ni is the number of the card (1, 2, 3, 4, 5, 6, 7, 10, 11 or 12), and Si is its suit (espada, basto, copa or oro). You may assume that the three cards are different. The last line of the input contains three times the number −1 and an asterisk, with the six values separated by single spaces, and should not be processed as a test case.

Output

For each test case output a single line with an integer representing the maximum increase in the envido score that can be obtained by replacing exactly one card of the input hand.
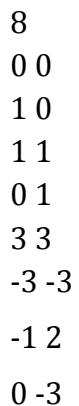
Example:

Input

12 espada 10 basto 11 basto
7 espada 1 oro 2 oro
7 espada 1 oro 6 espada
-1 * -1 * -1 *

Output

7
10
0

Question: - 5

There is a very large field, colored white and divided into squares. There is a coordinate system attached to the field: y-axis goes from south to north and x-axis goes from west to east. Sides of the squares are parallel to the axes. There is a robot in the (0, 0) square. Robot starts to move in spiral as depicted. First it moves one step east and one step north. Then two steps west and two steps south. Then three steps east and three steps north, four steps west and four steps south and so on. It moves to a new square with each step. As it moves it counts squares it passes and, if the number of the square is the prime number, then the robot fills this square with black color. The (0, 0) square has the number 0. Given the coordinates of a square you are to calculate the distance from this square to the nearest to it black square. For two squares with coordinates (x1, y1) and (x2, y2) the distance between those squares is |x1-x2|+|y1-y2|.

Input

Input file consists of a set of tests. The first line of the file is number T – the number of tests (T <= 500). Following T lines contains two integers each separated with a space: x and y – the coordinates of the square (-2000001 < x < 2000001, -2000001 < y < 2000001).

Output

For each coordinate pair in the input file you are to output the distance between this square and the nearest to it black square.

Example:

Input

8
0 0
1 0
1 1
0 1
3 3
-3 -3
-1 2
0 -3

Output

1
1
0
0
1
1
2
2