Lab Report on CSE 1200

Submitted by

Name: Md. Shabir Khan Akash.

Class: 1st year, even semester.

Dept.: Department of CSE

Roll: 1603108

Section: B

Submitted to

Md. Asifur Rahman

Lecturer

Department of CSE, RUET

UVa Online Judge

Problem no. p272

TEX is a typesetting language developed by Donald Knuth. It takes source text together with a few typesetting instructions and produces, one hopes, a beautiful document. Beautiful documents use “and ” to delimit quotations, rather than the mundane " which is what is provided by most keyboards. Keyboards typically do not have an oriented double-quote, but they do have a left-single-quote ` and a right-single-quote '. Check your keyboard now to locate the left-single-quote key ` (sometimes called the “backquote key”) and the right-single-quote key ' (sometimes called the “apostrophe” or just “quote”). Be careful not to confuse the left-single-quote ` with the “backslash” key \. TEX lets the user type two left-single-quotes `` to create a left-double-quote “ and two right-single-quotes " to create a right-double-quote ”. Most typists, however, are accustomed to delimiting their quotations with the un-oriented double-quote ".

If the source contained
"To be or not to be," quoth the bard, "that is the question."
then the typeset document produced by TEX would not contain the desired form:
“To be or not to be,” quoth the bard, “that is the question.”
In order to produce the desired form, the source file must contain the sequence:
``To be or not to be," quoth the bard, ``that is the question."

You are to write a program which converts text containing double-quote (") characters into text that is identical except that double-quotes have been replaced by the two-character sequences required by TEX for delimiting quotations with oriented double-quotes. The double-quote (") characters should be replaced appropriately by either `` if the " opens a quotation and by " if the " closes a quotation. Notice that the question of nested quotations does not arise: The first " must be replaced by ``, the next by ", the next by ``, the next by ", the next by ``, the next by ", and so on.

Input

Input will consist of several lines of text containing an even number of double-quote (") characters. Input is ended with an end-of-file character.

Output

The text must be output exactly as it was input except that:
• the first " in each pair is replaced by two ` characters: `` and
• the second " in each pair is replaced by two ' characters: ".

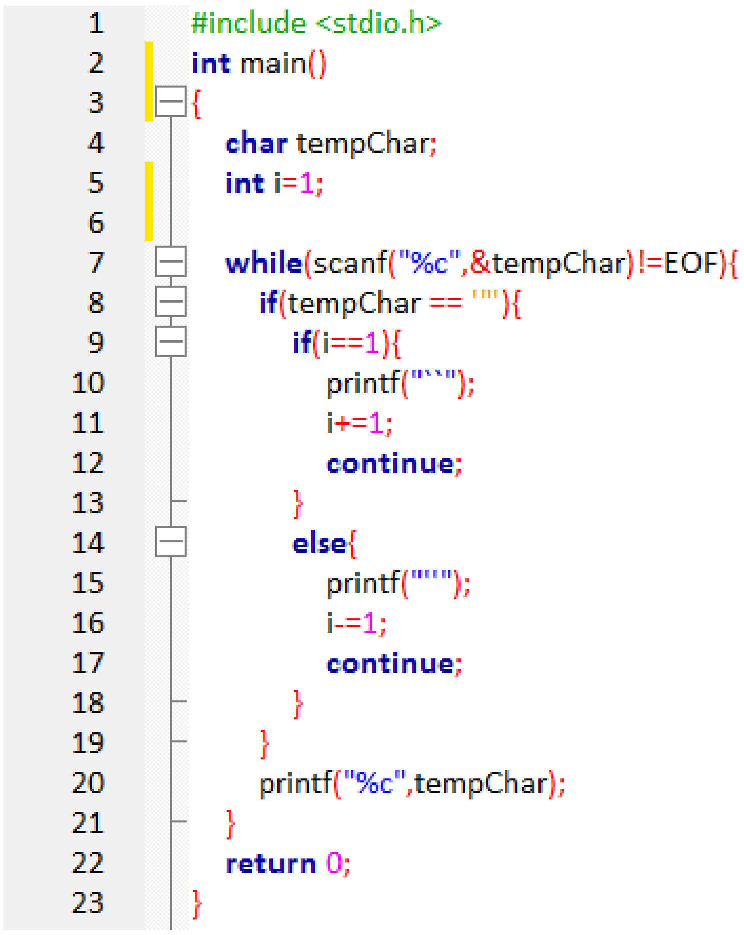
Sample Input

"To be or not to be," quoth the Bard, "that is the question".
The programming contestant replied: "I must disagree. To `C' or not to `C', that is The Question!"

Sample Output

``To be or not to be," quoth the Bard, ``that is the question".
The programming contestant replied: ``I must disagree.
To `C' or not to `C', that is The Question!"

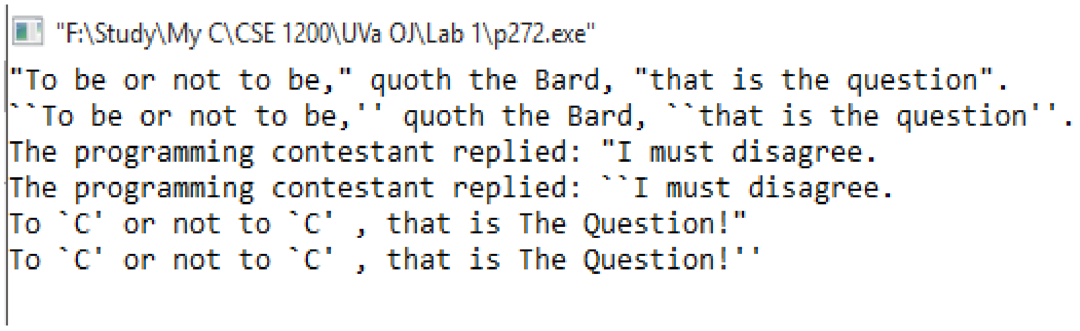
Code



```c
#include <stdio.h>
int main()
{
    char tempChar;
    int i=1;

    while(scanf("%c",&tempChar)!=EOF){
        if(tempChar == '"'){
            if(i==1){
                printf("``");
                i+=1;
                continue;
            }
            else{
                printf("''");
                i-=1;
                continue;
            }
        }
        printf("%c",tempChar);
    }
    return 0;
}
```

Output



```
"F:\Study\My C\CSE 1200\UVa OJ\Lab 1\p272.exe"
"To be or not to be," quoth the Bard, "that is the question".
``To be or not to be,'' quoth the Bard, ``that is the question''.
The programming contestant replied: "I must disagree.
The programming contestant replied: ``I must disagree.
To `C' or not to `C' , that is The Question!"
To `C' or not to `C' , that is The Question!''
```

Analysis of solution

In the above program, a while loop was used to input character from the user until it reaches to EOF. Then a condition was applied to check if the input character was double quotation (“) or not if so than another condition occurred checking the value of i was 1 or not if so then it printed (`) symbol replacing the first double quotation mark (“) and I was incremented by 1 and continued to do that operation again and as i was incremented by 1 then after next check it went to else section to replace the 2nd quotation mark (“) with two single (‘) marks and the value of i was decremented by 1 and continued taking character inputs from the user.

UVa Online Judge

Problem no. p1124

It's hard to construct a problem that's so easy that everyone will get it, yet still difficult enough to be worthy of some respect. Usually, we err on one side or the other. How simple can a problem really be?

Here, as in Celebrity Jepoardy, questions and answers are a bit confused, and, because the participants are celebrities, there's a real need to make the challenges simple. Your program needs to prepare a question to be solved | an equation to be solved | given the answer. Specically, you have to write a program which ends the simplest possible equation to be solved given the answer, considering all possible equations using the standard mathematical symbols in the usual manner. In this context, simplest can be defined unambiguously several different ways leading to the same path of resolution. For now, and the equation whose transformation into the desired answer requires the least effort.

For example, given the answer X = 2, you might create the equation 9   X = 7. Alternately, you could build the system X > 0; X2= 4. These may not be the simplest possible equations. Solving these mind-scratchers might be hard for a celebrity.

Input

Each input line contains a solution in the form < symbol > = < value >

Output

For each input line, print the simplest system of equations which would to lead to the provided solution,
respecting the use of space exactly as in the input.

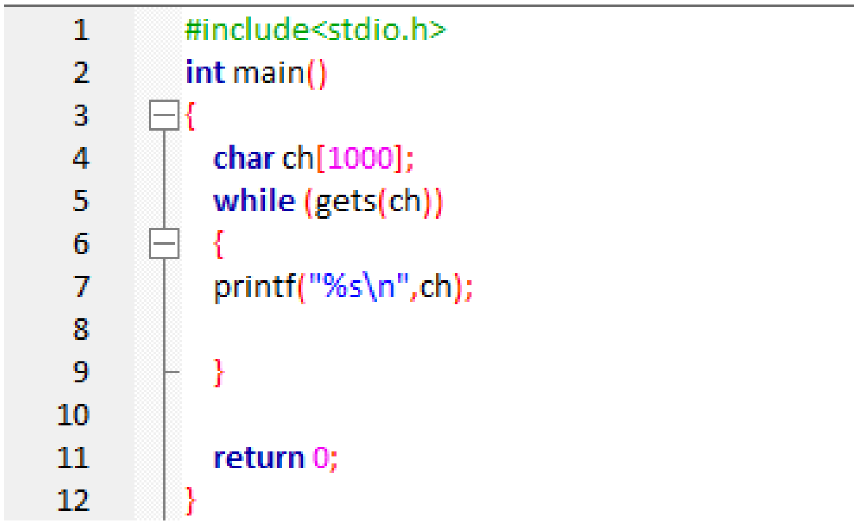
Sample Input

```
Y = 3
X=9
```

Sample Output

```
Y = 3
X=9
```

Code

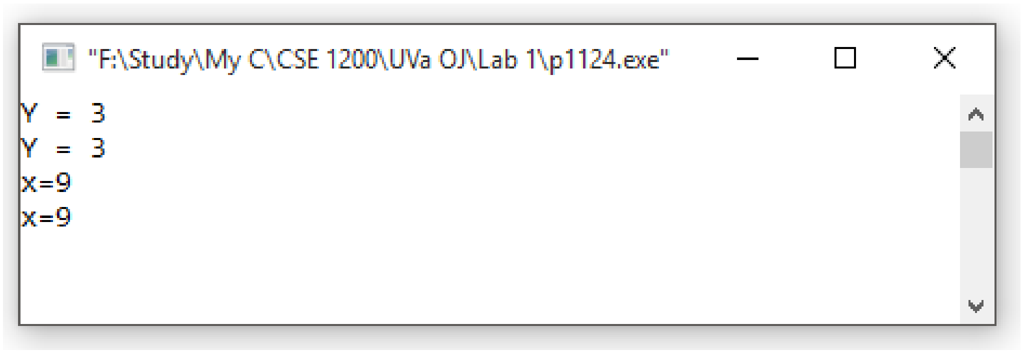


```c
#include<stdio.h>
int main()
{
    char ch[1000];
    while (gets(ch))
    {
    printf("%s\n",ch);

    }

    return 0;
}
```

Output



"F:\Study\My C\CSE 1200\UVa OJ\Lab 1\p1124.exe"

```
Y  =  3
Y  =  3
x=9
x=9
```

Analysis of Solution

In this above code, the possible simplest form of an equation to be shown in output. So in the body of the program a while loop was used to input elements as an element of a string ‘ch’ until input file reaches to EOF and simply in the loop, that input string was printed on the console as a string because symbol and value is the simplest form of a equation term.

UVa Online Judge

Problem no. p11044

The Loch Ness Monsteris a mysterious and unidentified animal said to inhabit Loch Ness, a large deep freshwater loch near the city of Inverness in northern Scotland. Nessie is usually categorized as a type of lake monster.
http://en.wikipedia.org/wiki/Loch Ness Monster
In July 2003, the BBC reported an extensive investigation of Loch Ness by a BBC team, using 600 separate sonar beams, found no trace of any \sea monster" (i.e., any large animal, known or unknown) in the loch. The BBC team concluded that Nessie does not exist. Now we want to repeat the experiment.

Given a grid of n rows and m columns representing the loch, 6 n, m 10000, and the minimum number s of sonar beams you must put in the square such that we can control every position in the grid, with the following conditions:

• one sonar occupies one position in the grid; the sonar
beam controls its own cell and the contiguous cells;
• the border cells do not need to be controlled, because
Nessy cannot hide there (she is too big).

For example,
where X represents a sonar, and the shaded cells are controlled by their sonar beams; the last figure gives us a solution.

Input

The first line of the input contains an integer, t, indicating the number of test cases. For each test case, there is a line with two numbers separated by blanks, 6 <= n, m <= 10000, that is, the size of the grid (n rows and m columns).

Output

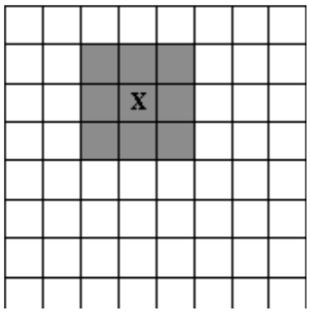
For each test case, the output should consist of one line showing the minimum number of sonars that verifies the conditions above.

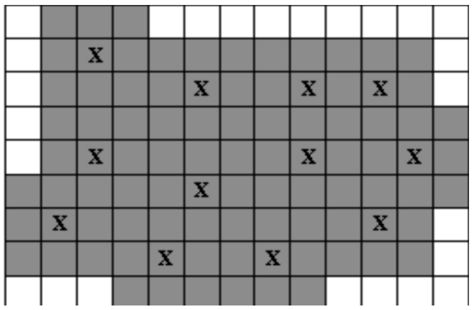
Sample Input

```
3
6 6
7 7
9 13
```

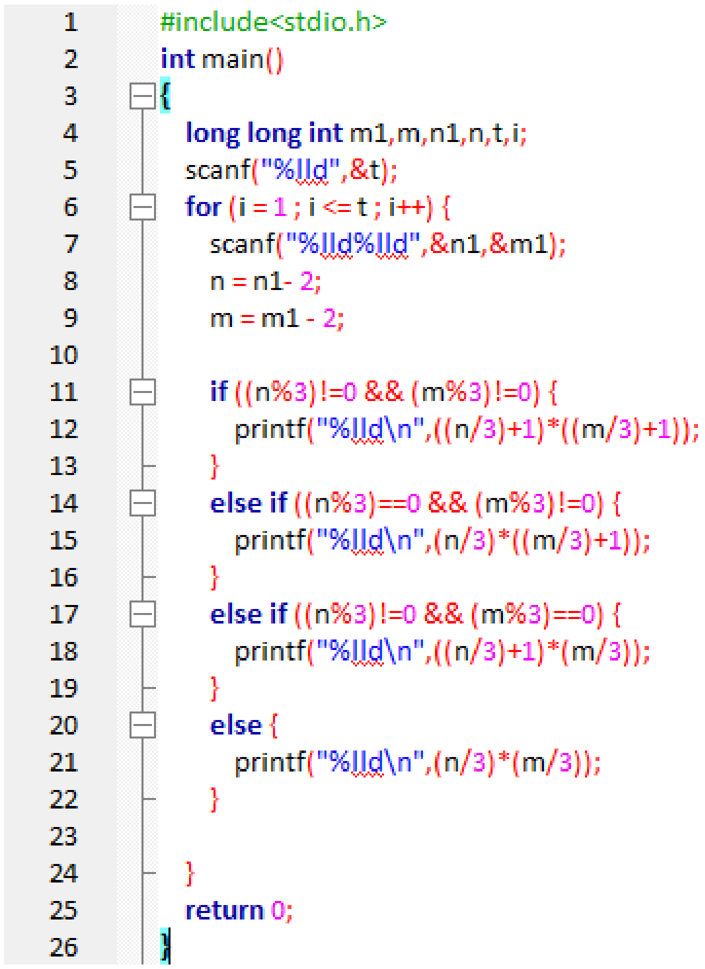
Sample Output

```
4
4
12
```





Code

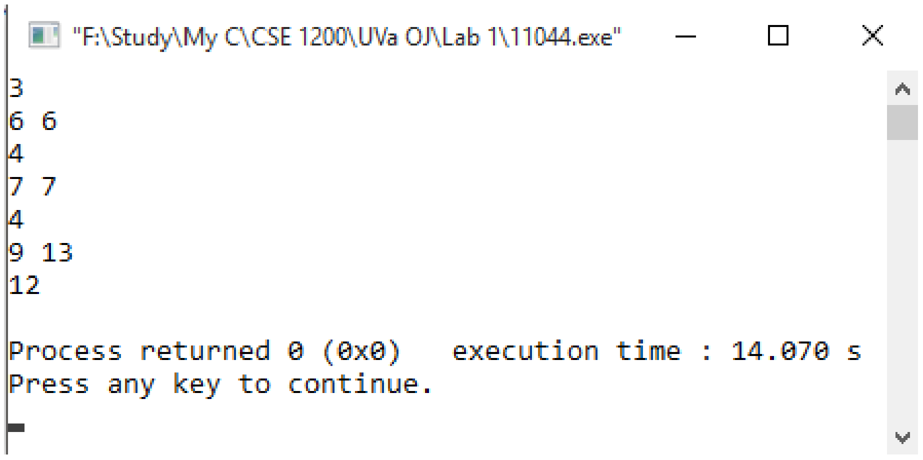


```c
#include<stdio.h>
int main()
{
    long long int m1,m,n1,n,t,i;
    scanf("%lld",&t);
    for (i = 1 ; i <= t ; i++) {
        scanf("%lld%lld",&n1,&m1);
        n = n1- 2;
        m = m1 - 2;

        if ((n%3)!=0 && (m%3)!=0) {
            printf("%lld\n",((n/3)+1)*((m/3)+1));
        }
        else if ((n%3)==0 && (m%3)!=0) {
            printf("%lld\n",(n/3)*((m/3)+1));
        }
        else if ((n%3)!=0 && (m%3)==0) {
            printf("%lld\n",((n/3)+1)*(m/3));
        }
        else {
            printf("%lld\n",(n/3)*(m/3));
        }

    }
    return 0;
}
```

Output



"F:\Study\My C\CSE 1200\UVa OJ\Lab 1\11044.exe"

```
3
6 6
4
7 7
4
9 13
12

Process returned 0 (0x0)   execution time : 14.070 s
Press any key to continue.
```

Analysis of Solution

In the above program, firstly a for loop continues to works till the number of test cases reached and then 2 was subtracted from the value of both n1 (row) and m1 (column) as the border of the cell doesn’t count because the monster can’t hide in there. Then it was checked whether the total rows and total columns are divisible by 3 or not , if not then the rows were divided by 3 and added with 1 to and same story goes with column also and after that both of them were multiplied to get the area where the monster can hide out. If both rows and columns are divisible by 3 then same above operation will occur but the added 1 will not be present in both row and column portion as they are already divisible by 3. Then if row is not divisible by 3 but column is divisible then only (row/3) will be added with 1 and same operation goes for column also.

UVa Online Judge

Problem no. p10550

Now that you're back to school for another term, you need to remember how to work the combination lock on your locker. A common design is that of the Master Brand, shown at right. The lock has a dial with 40 calibration marks numbered 0 to 39. A combination consists of 3 of these numbers; for example:
15-25-8. To open the lock, the following steps are taken:
- turn the dial clockwise 2 full turns
- stop at the first number of the combination
- turn the dial counter-clockwise 1 full turn
- continue turning counter-clockwise until the 2nd number is reached
- turn the dial clockwise again until the 3rd number is reached
- pull the shank and the lock will open.

Given the initial position of the dial and the combination for the lock, how many degrees is the dial rotated in total (clockwise plus counter-clockwise) in opening the lock?

Input

Input consists of several test cases. For each case there is a line of input containing 4 numbers between 0 and 39. The first number is the position of the dial. The next three numbers are the combination. Consecutive numbers in the combination will be distinct. A line containing `0 0 0 0' follows the last case.

Output

For each case, print a line with a single integer: the number of degrees that the dial must be turned to open the lock.

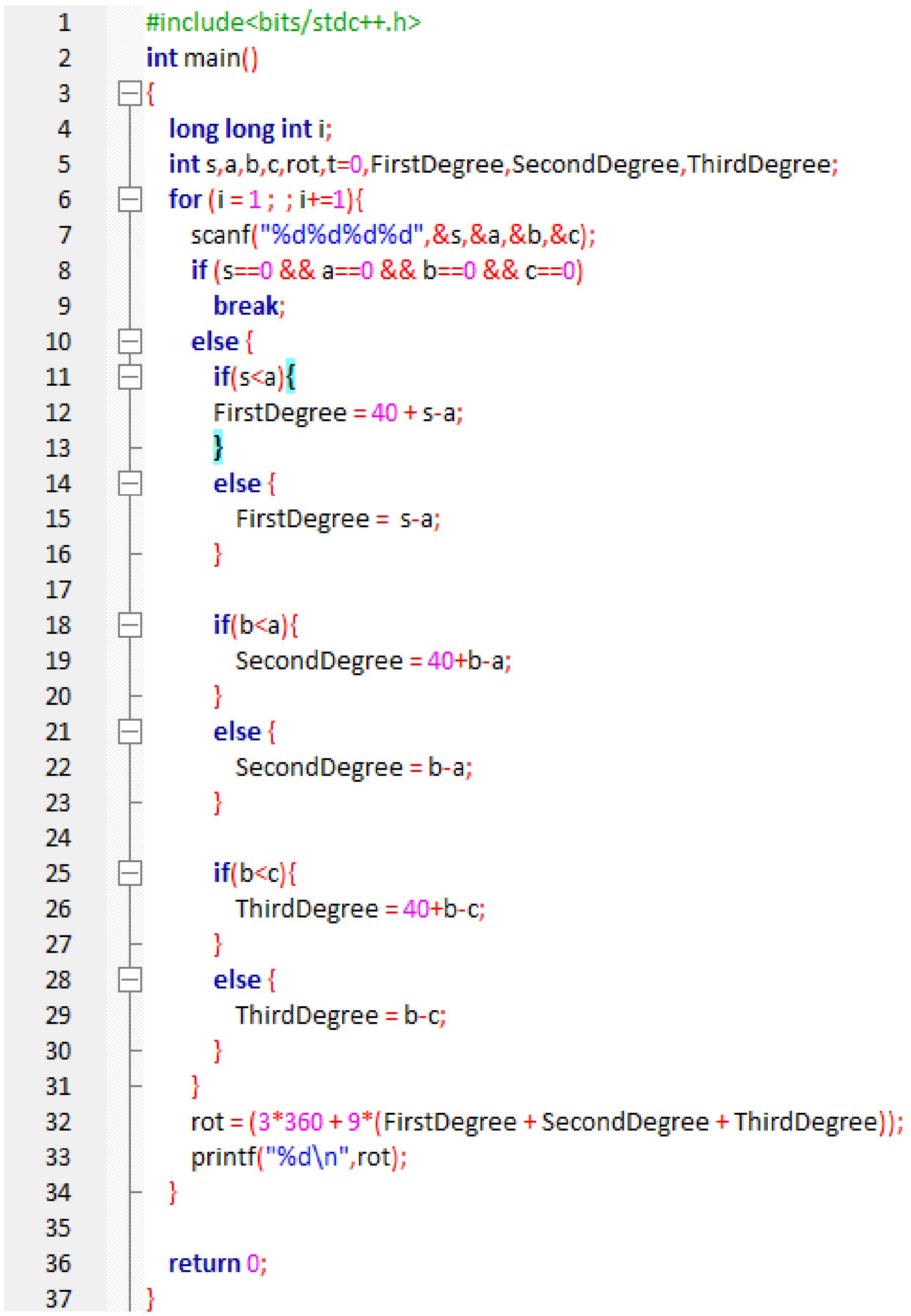
Sample Input

```
0 30 0 30
5 35 5 35
0 20 0 20
7 27 7 27
0 10 0 10
9 19 9 19
0 0 0 0
```

Sample Output

```
1350
1350
1620
1620
1890
1890
```

Code



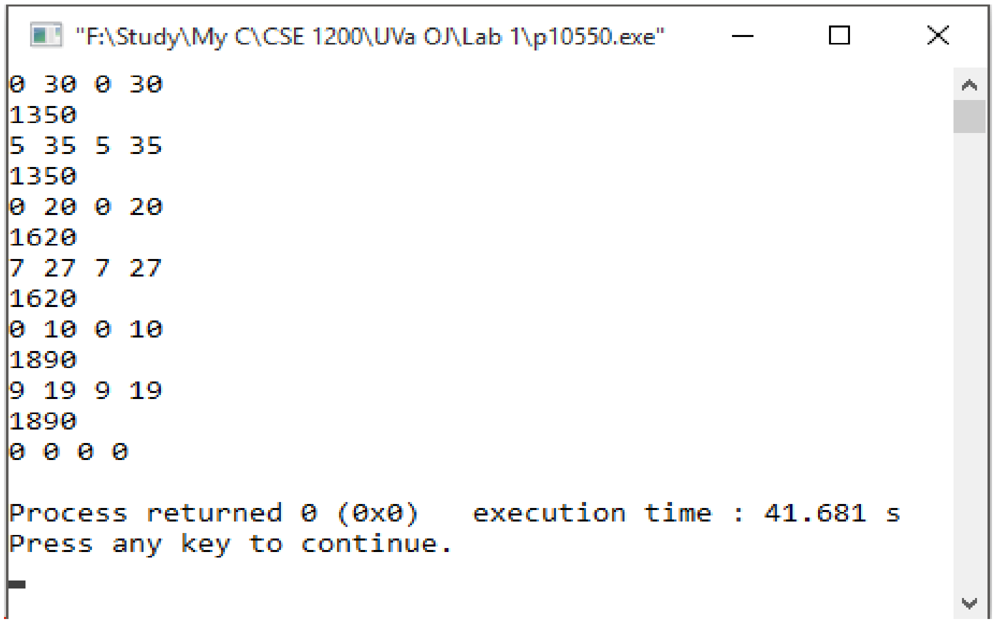
```
#include<bits/stdc++.h>
int main()
{
  long long int i;
  int s,a,b,c,rot,t=0,FirstDegree,SecondDegree,ThirdDegree;
  for (i = 1 ;  ; i+=1){
    scanf("%d%d%d%d",&s,&a,&b,&c);
    if (s==0 && a==0 && b==0 && c==0)
      break;
    else {
      if(s<a){
      FirstDegree = 40 + s-a;
      }
      else {
        FirstDegree =  s-a;
      }

      if(b<a){
        SecondDegree = 40+b-a;
      }
      else {
        SecondDegree = b-a;
      }

      if(b<c){
        ThirdDegree = 40+b-c;
      }
      else {
        ThirdDegree = b-c;
      }
    }
    rot = (3*360 + 9*(FirstDegree + SecondDegree + ThirdDegree));
    printf("%d\n",rot);
  }

  return 0;
}
```

Output



"F:\Study\My C\CSE 1200\UVa OJ\Lab 1\p10550.exe"

```
0 30 0 30
1350
5 35 5 35
1350
0 20 0 20
1620
7 27 7 27
1620
0 10 0 10
1890
9 19 9 19
1890
0 0 0 0

Process returned 0 (0x0)   execution time : 41.681 s
Press any key to continue.
```

Analysis of Solution

In the above program, in the input if s (the first number) is less then a (the second number) then s was added with 40 as the lock has 40 calibration marks and then a was subtracted from that summation, otherwise no 40 would be added. Similarly same checking was occurred between b (the third number) and a then the operation was also same. After that another same of this checking was done between b and c (the fourth number) and same operation was performed again. To get the total rotation all of the above summation was multiplied by 9 as each calibration number was 9 degree away from its very next calibration number and that was added with 1080 as it is the total amount of default rotation for clockwise and anticlockwise rotation of the dial. After this total summation is the required output.

UVa Online Judge

Problem no. p11172

Some operators checks about the relationship between two values and these operators are called relational operators. Given two numerical values your job is just to and out the relationship between them that is (i) First one is greater than the second (ii) First one is less than the second or (iii) First and second one is equal.

Input

First line of the input file is an integer t (t < 15) which denotes how many sets of inputs are there. Each of the next t lines contain two integers a and b (|a|,|b| < 1000000001).

Output

For each line of input produce one line of output. This line contains any one of the relational operators
`>', `<' or `=', which indicates the relation that is appropriate for the given two numbers.

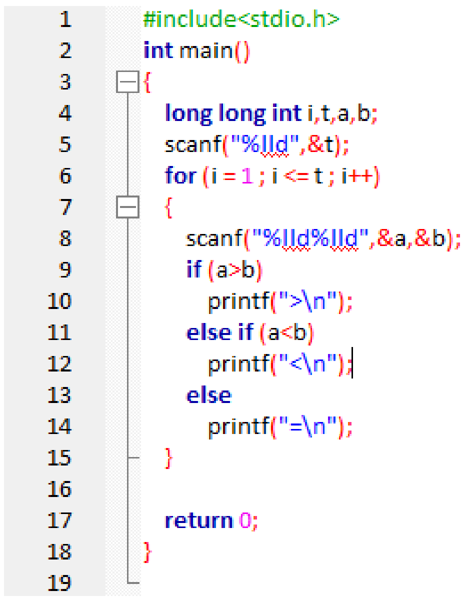
Sample Input

```
3
10 20
20 10
10 10
```

Sample Output

```
<
>
=
```

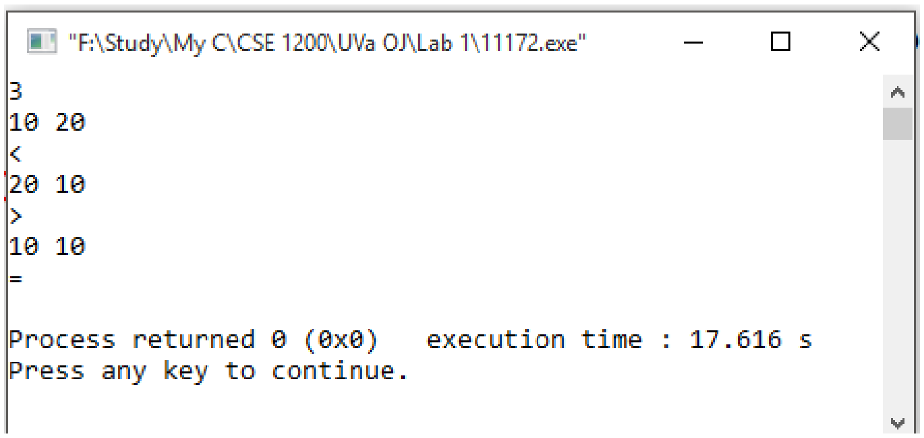
Code



```
#include<stdio.h>
int main()
{
   long long int i,t,a,b;
   scanf("%lld",&t);
   for (i = 1 ; i <= t ; i++)
   {
      scanf("%lld%lld",&a,&b);
      if (a>b)
         printf(">\n");
      else if (a<b)
         printf("<\n");
      else
         printf("=\n");
   }

   return 0;
}
```

Output


"F:\Study\My C\CSE 1200\UVa OJ\Lab 1\11172.exe"

```
3
10 20
<
20 10
>
10 10
=

Process returned 0 (0x0)   execution time : 17.616 s
Press any key to continue.
```

Analysis of Solution

In this above program, two numbers were taken as input and just simple conditional logic applied on them just to check whether first number is greater or the second or both are equal. Depending on the number the required sign was printed on the output.

Codeforces Online Judge

Problem no. 1A

Theatre Square

Theatre Square in the capital city of Berland has a rectangular shape with the size $n \times m$ meters. On the occasion of the city's anniversary, a decision was taken to pave the Square with square granite flagstones. Each flagstone is of the size $a \times a$.
What is the least number of flagstones needed to pave the Square? It's allowed to cover the surface larger than the Theatre Square, but the Square has to be covered. It's not allowed to break the flagstones. The sides of flagstones should be parallel to the sides of the Square.

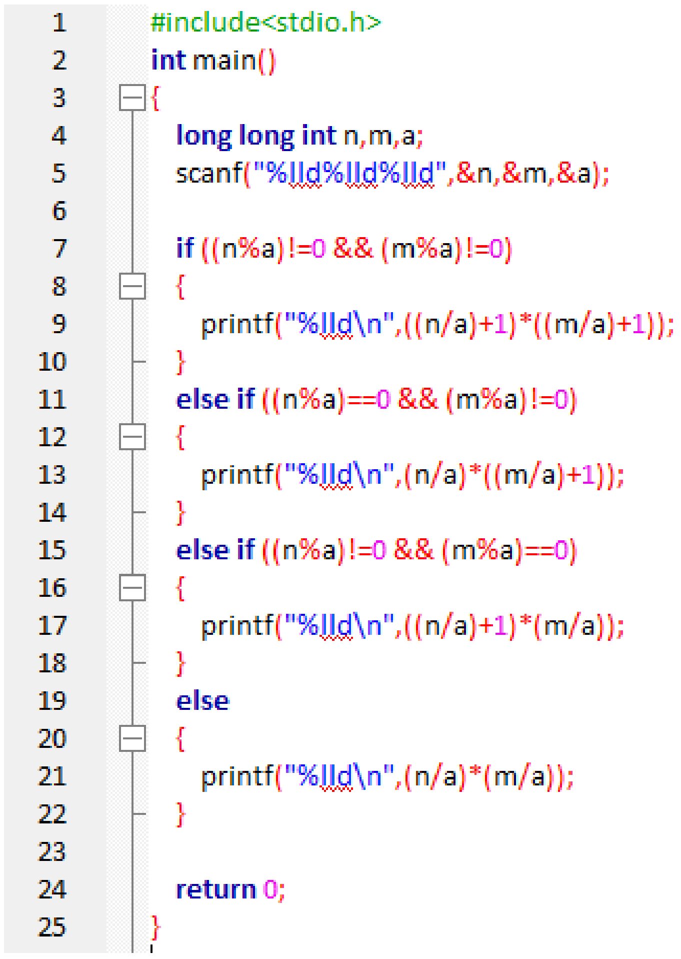
Input

The input contains three positive integer numbers in the first line: n, m and a ($1 \le n, m, a \le 10^9$).

Output

Write the needed number of flagstones.

Code

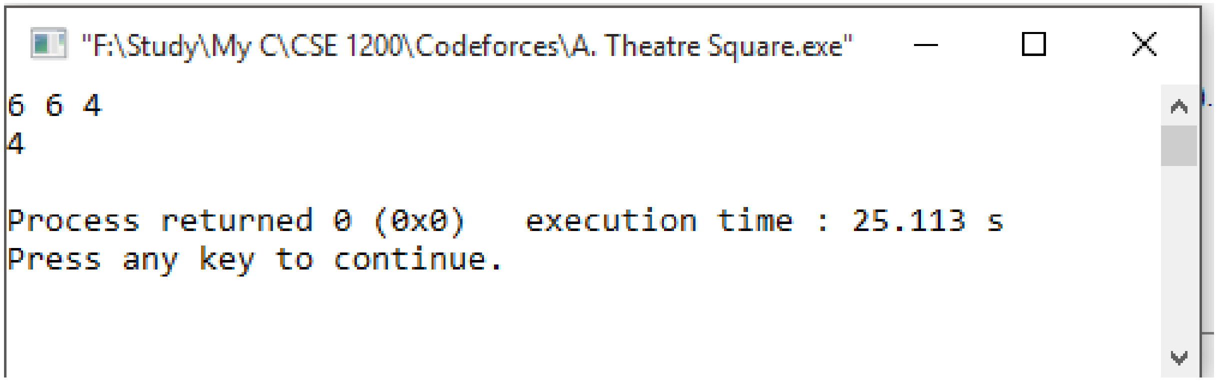


```c
#include<stdio.h>
int main()
{
    long long int n,m,a;
    scanf("%lld%lld%lld",&n,&m,&a);

    if ((n%a)!=0 && (m%a)!=0)
    {
        printf("%lld\n",((n/a)+1)*((m/a)+1));
    }
    else if ((n%a)==0 && (m%a)!=0)
    {
        printf("%lld\n",(n/a)*((m/a)+1));
    }
    else if ((n%a)!=0 && (m%a)==0)
    {
        printf("%lld\n",((n/a)+1)*(m/a));
    }
    else
    {
        printf("%lld\n",(n/a)*(m/a));
    }

    return 0;
}
```

Output



"F:\Study\My C\CSE 1200\Codeforces\A. Theatre Square.exe"

```
6 6 4
4

Process returned 0 (0x0)   execution time : 25.113 s
Press any key to continue.
```

Analysis of Solution

In the above program, it was checked whether the total rows and total columns are divisible by a (size of flagstone) or not, if not then the rows were divided by a and added with 1 to and same story goes with column also and after that both of them were multiplied to get the area where the square city can be covered with the flagstones completely. If both rows and columns are divisible by a then same above operation will occur but the added 1 will not be present in both row and column portion as they are already divisible by a. Then if row is not divisible by a but column is divisible then only (row/a) will be added with 1 and same operation goes for column also.