Indian National Olympiad in Informatics, 2003

Time: 3 hours *30 April, 2003*

Instructions

(a) You will have to return this question paper at the end of the examination with relevant parts filled out.

(b) There are two questions. You have to write working programs in Pascal, C or C++ to solve each of these questions. You must compile your program yourself and produce an executable file (EXE file).

At the end of each question, there is a space to indicate the location of the source code and the executable file for your solution. Please fill up this information without fail.

(c) All input for your programs will come from the keyboard. All output from your programs should be written to the screen.

Question 1 Chambers in a castle

The floor plan of a castle indicates where the walls are. The outer boundary of the castle is always an unbroken wall. The castle has no doors, only openings in the walls that lead from one room to another. A *chamber* is a collection of rooms that are connected to each other by gaps in the walls. The problem is to:

(a) Count the number of chambers in the castle.

(b) Calculate the area of the largest chamber.

To simplify the problem, the floor of the castle is divided into square tiles. All walls lie at tile boundaries. The area of a chamber is specified in terms of the number of tiles inside the room (without counting the tiles occupied by the surrounding walls).

Figure 1 is an example of a floor plan. Each square is a tile. White tiles represent open space, while black tiles represent walls. In this castle, there are four chambers. The largest chamber has an area of 58.

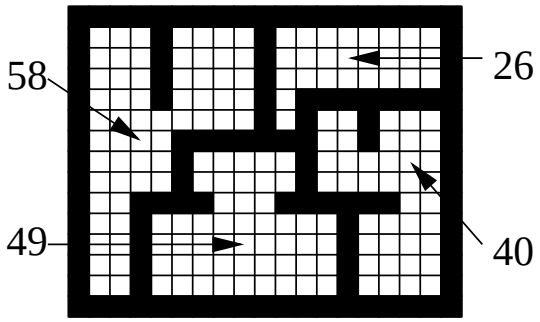

Figure 1

Input format

The first line of input will have two integers `M` and `N`, with $0 < \texttt{M} \leq 500$ and $0 < \texttt{N} \leq 500$. `M` represents the number of rows and `N` the number of columns in the floor plan.

This will be followed by `M` lines of input. Each of these `M` lines will be a sequence of 0's and 1's of length `N`, separated by spaces. A 0 represents an open tile (a white square), while a 1 represents a tile with a wall (a black square). The outer boundary of the castle will always consist of an unbroken wall.

The input corresponding to the floor plan in Figure 1 is given in Figure 2.

```
15 19
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 1 0 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1
1 0 0 0 0 1 1 1 1 1 1 1 0 0 1 0 0 0 1
1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1
1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1
1 0 0 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 1
1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Figure 2

Output format

The output of your program should be exactly two lines, each line consisting of a single number. The first line should be the number of chambers. The second line should be the size of the largest chamber.

The correct output for the example in Figures 1 and 2 is shown below.

```
4
58
```

Note: Your program should not print anything other than these two numbers. Please remove all diagnostic print statements before making your final submission. A program with extraneous output will be treated as incorrect!

Question 2 Nearest fraction

Let X be the set of all fractions in reduced form lying strictly below 0 and 1 whose denominator is less than or equal to 99. In other words,

$$\frac{n}{d} \text{ belongs to } X \text{ provided } 0 < \frac{n}{d} < 1 \text{ and } d \leq 99 \text{ and } \gcd(n, d) = 1,$$

where $\gcd(x, y)$ denotes the *greatest common divisor* (or *highest common factor*) of x and y.

For instance, X includes fractions such as 1/3, 11/31 and 24/37 and excludes fractions such as 4/10, 30/70 (both not in reduced form) and 2/101 (denominator too large).

The problem

Given an arbitrary fraction a/b in reduced form whose denominator b is larger than 99, we want to find the pair of fractions in X that are closest to a/b.

In other words, we want to identify fractions u/v and x/y in X such that $u/v < a/b < x/y$ with the property that there is no fraction u'/v' in X such that $u/v < u'/v' < a/b$ and there is no fraction x'/y' in X such that $a/b < x'/y' < x/y$.

For instance, if a/b is 2/101, then $u/v = 1/51$ and $x/y = 1/50$. Here is another example—if $a/b = 322/479$, then $u/v = 41/61$ and $x/y = 39/58$.

Input format

Each test input will consist of a sequence of values a/b for which you have to find the nearest fractions in X. The input is given as follows.

The first line is an integer `M`, $0 < \texttt{M} \leq 500$, indicating the number of fractions in this input test sequence. This is followed by `M` lines of input, each containing a pair of integers `N` and `D` separated by a space, representing the numerator and denominator of the input fraction, respectively. You are guaranteed that $\texttt{D} \geq 100$ and $\gcd(\texttt{N}, \texttt{D}) = 1$.

Here is what the input would look like if the sequence consisted of the two examples 2/101 and 322/479 discussed earlier.

```
2
2 101
322 479
```

Output format

For each input fraction n/d, you have to print out a line containing the values u/v and x/y nearest to n/d in X, where $u/v < n/d < x/y$. Print out u, v, x and y as four integers, in that order, on a single line, separated by spaces. Thus, your output will consist of `M` lines overall, each containing 4 integers.

The correct output for the earlier sample input is shown below.

```
1 51 1 50
41 61 39 58
```

Note: Your program should not print anything other than these `M` lines, each consisting of 4 numbers. Please remove all diagnostic print statements before making your final submission. A program with extraneous output will be treated as incorrect!