

# British Informatics Olympiad Final

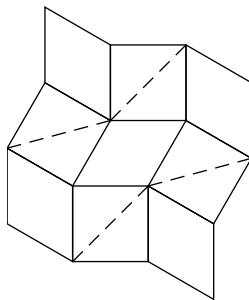
11–13 April, 2003

Sponsored by Lionhead Studios

## Wonky Widgets

Hanging behind Mr Widget senior, in his office at *Widget, Whatsit & Doodah* (est. 1862), is a small plaque. Worn with age, for it dates back to the time of Widget grand-senior, it bears the simple legend **Wonky Widgets = Weakened Widgets**. This refers to the time-honoured way of making widgets, using an internal grid of scaffolding, which was liable to deform unless properly braced.

A traditional scaffold is a grid of unit squares, made from unit length rods which are linked by complicated joints at their endpoints. These joints are flexible so the grid can be deformed. For example, the following pictures (ignore the dotted lines for now) shows a deformed  $3 \times 3$  grid. Note how all the (solid) lines have the same length.



Individual squares can be made rigid by adding a brace across their diagonal (shown by the dotted lines in the above picture). Squares might be rigid without braces, if they are forced that way by other rigid squares. For example, if a brace was added to the middle square in the picture, it would force the entire grid (including the corner squares without braces) to be rigid. Since braces add to the cost of the widget, it is desirable to minimise their number.

Since the grid is made of unit length rods (and hence deformed squares are parallelograms) all the vertical rods in a given row must be parallel. Similarly, all the horizontal rods in a given column must be parallel. A brace in row  $i$  and column  $j$  forces all the vertical lines in row  $i$  to be at right angles to the horizontal lines in column  $j$ . The entire grid is rigid if all the vertical lines are parallel to each other (similarly for the horizontal lines).

Write a program that, given a list of braces already on a widget's scaffolding, determines the *minimum* number of braces that must be added to the existing design to prevent the widget becoming wonky. The first line of the input will contain two integers,  $r$  then  $c$  ( $1 \leq r, c \leq 100$ ), indicating the number of rows and columns in the widget. This will be followed by a list of braced squares, one brace on each line, each given by its row followed by its column; Rows will be numbered from 1 to  $r$ , and columns from 1 to  $c$ . The list of braces will be terminated by the line  $-1 -1$ . No brace will be given more than once.

You should output a list of the braces to be added, identifying each brace by its row then column. Each brace should appear on a separate line and no brace should appear more than once.

### Sample Input

```
3 3
3 2
2 1
2 3
3 3
2 2
-1 -1
```

### Sample Output

```
1 2
```