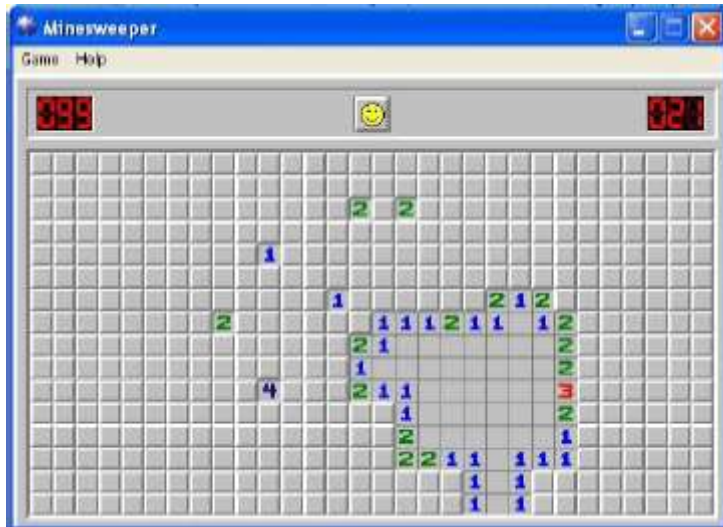


## ICS 4U1 - Assignment #3

### 1) Minesweeper

Minesweeper is a game available with the Windows™ operating system. The object of Minesweeper is to locate all the mines as quickly as possible without uncovering any of them. You can uncover a square by clicking it. If you uncover a mine, you lose the game. If a number appears on a square, it indicates how many mines are in the eight squares that surround the numbered one.



Sometimes clicking on a square, that has no mines surrounding it, reveals many squares. It will reveal all of the squares surrounding it until it reaches squares that have mines surrounding it.

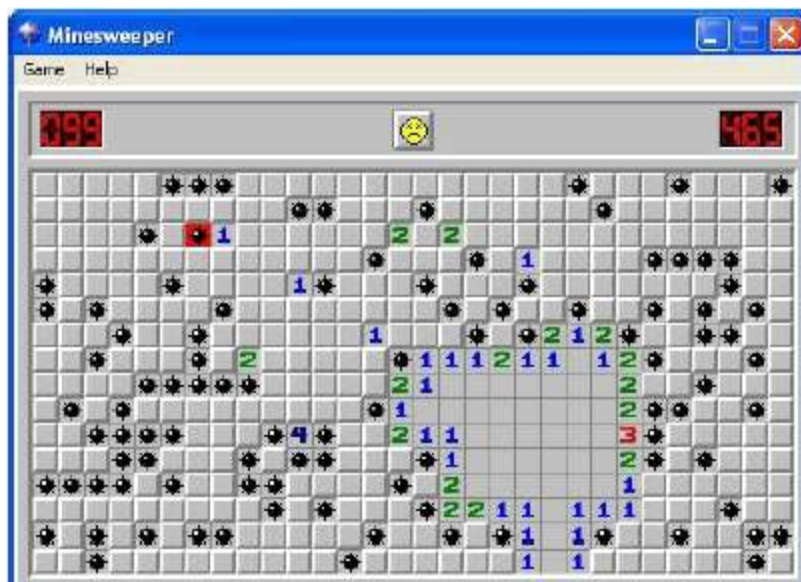
Your job is to write a program to determine the result of clicking on a square during a Minesweeper game. The input file (**DATA.txt**) will contain twenty-one lines of data. The first sixteen lines will contain thirty characters each. Each character will be either an 'X', that represents a square

with a mine or a period (.) that will represent a square without a mine. The last five lines will each contain two integers, r and c, that represent the row and column of the square that is clicked. Assume the top left square of the minefield to be at row one and column one. The output to the screen will contain the results of clicking on the square. For sake of simplicity, assume that each of the clicks, from the input file, is the first click of a new game (i.e. the game board won't be changed).

If the square contains a mine, output "MINE - YOU LOSE".

If the square does not contain a mine, but there are some mines surrounding it, output "NO MINE - # SURROUNDING IT", where # is the number of mines surrounding it.

If the square does not contain a mine and there are no mines surrounding it, output "NO MINE - # SQUARES REVEALED", where # is the number of squares revealed.



### Sample Input

```

.....XXX.....X...X...X
.....XX...X.....X.....
...X.X.....
.....X...X.....XXXX..
X...X...X...X...X.....X..
X.X...X.....X.X.X.X.X.X.
...X..X.....X.X...X..XX..
..X...X.....X.....X...X.
....XXXXX.....X...
.X.X.....X.....XX..X.
..XXXX...X.X.....X.....
...XX...X.XX...X.....X.X...
XXXXX.X..XX...X.....
.....X.X...X.....X..
X.X.X.XX.....X..X.X...X..XX
..X.....X.....X.
1 6
3 8
6 2
10 18
16 29

```

### Sample Output

```

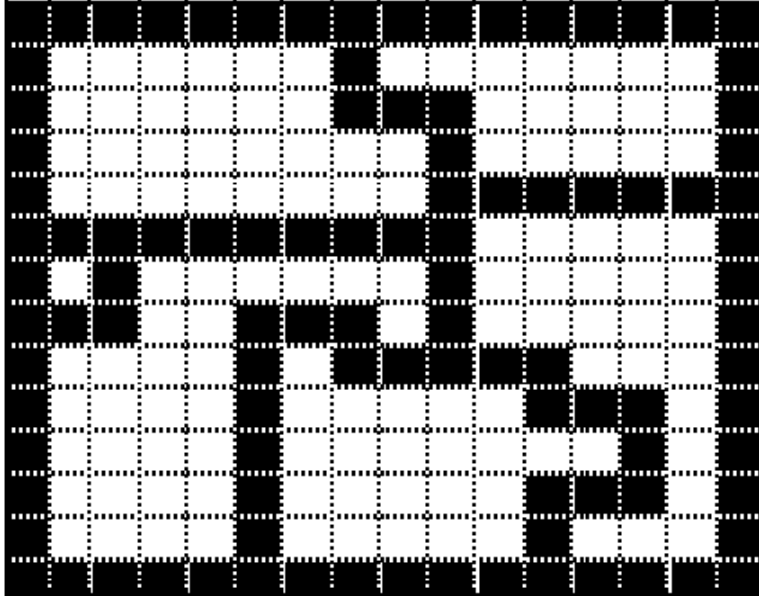
MINE at (1,6) - YOU LOSE
NO MINE at (3,8) - 1 SURROUNDING IT
NO MINE at (6,2) - 3 SURROUNDING IT
NO MINE at (10,18) - 72 SQUARES REVEALED
MINE at (16,29)- YOU LOSE

```

## 2) Floor Plan

The floor plan of a house shows rooms separated by walls. This floor plan can be transferred to a grid using the character “I” for walls and “.” for room space. Doorways are not shown. Each “I” or “.” character occupies one square metre.

In this diagram there are six rooms.



You have been given the floor plan of a house and a supply of hardwood flooring. You are to determine how many rooms will have the flooring installed if you start installing the floor in the largest room first and move to the next largest room, and so on. You may not skip over any room, and you must stop when you do not have enough wood for the next room. Output the number of rooms that can have hardwood installed, and how many square metres of flooring are left over.

No room will be larger than 64 square metres.

The first line of the data file contains the number of square metres of flooring you have. The second line in the file contains an integer  $r$  from 1 – 25 that represents the number of rows in the grid. The third line contains an integer  $c$  from 1 – 25 that represents the number of columns in the grid. The remaining  $r$  lines contain  $c$  characters of grid data.

### Sample Input 1

```
125
14
16
IIIIIIIIIIIIIIIIII
I.....I.....I
I.....III.....I
I.....I.....I
I.....IIIIIII
IIIIIIIIII.....I
I.I.....I.....I
III..III.I.....I
I....I.IIIII...I
I....I.....III.I
I....I.....I.I
I....I.....III.I
I....I.....I...I
IIIIIIIIIIIIIIIIII
```

### Output for Sample Input 1

```
6 rooms, 3 square metre(s) left over
```

### Sample Input 2

```
13
2
3
.I.
.I.
```

### Output for Sample Input 2

```
2 rooms, 9 square metre(s) left over
```