

# NOI '07 P3 - Surrounding Battalions

---

**Time limit:** 120.0s    **Memory limit:** 256M

---

## National Olympiad in Informatics, China, 2007

Our army has just obtained intercepted information that the enemy is gathering forces in preparation for an attack on our important artillery research center. Since our forces are currently deployed to multiple battles, there is no way to concentrate large numbers of soldiers for going forth and providing support. Our commanders would like to make effective deployments in order to maximize our chances of winning, while minimizing casualties and losses.

The map of the artillery research center is an  $N \times M$  matrix. Each  $1 \times 1$  cell in the matrix represents a unit of area, and each cell is a neighbor of the cells to its top, bottom, left, and right. Each cell can be used in one of the three following ways:

1. as a military research center (denoted by the letter `O`);
2. as a station for a mechanized battalion (denoted by `#`); or
3. as open space (denoted by `.`).

Since area is limited, each  $1 \times 1$  cell cannot contain two or more units of mechanized battalions, otherwise mobility during battle will be greatly reduced.

Unfortunately, due to poor pre-war estimates, the deployment of our defense department appears very scattered. This makes it very easy for the enemy's surprise attack tactics to succeed. To ensure our absolute success, our forces have decided to take advantage of the few defense battalions we have to surround all important research areas, while also moving as few steps as possible. To so-called "surround" means to ensure that enemies coming from the border of the matrix cannot find any path to a cell with a research center without also undergoing the resistance of our mechanized battalions.

Due to the communication limitations of our army, the commanders' headquarters can only issue a command to a single battalion unit per unit of time (to go up, down, left, or right by 1 cell). Time is ticking, and headquarters hopes to finish the surrounding operation as soon as possible. And now, this critically important mission has been assigned to you to complete!

Note: during the surrounding process, a unit of battalion can temporarily occupy a research area. However, battalions cannot occupy the same cell as a research area after the surrounding operation has been completed. Additionally, during any given moment, two battalions may not occupy the same cell.

## Input Specification

---

There will be 10 files `surround1.in` to `surround10.in` that will be given to your program (through standard input). They can be downloaded here for you to study: [surround.zip](#).

For each of the test cases:

The first line will contain an integer between 1 and 10 inclusive, specifying the test case number (`surroundi.in` will have the integer  $i$  on the first line).

The second line will contain two integers  $N$  and  $M$ , the number of rows and columns in the matrix, respectively. The next  $N$  lines each contain  $M$  characters (either `.`, `0`, or `#`).

## Output Specification

The first line of output should contain an integer  $T$ , the time spent by your solution.

For the next  $T$  lines, each line should contain 4 integers  $x_1, y_1, x_2,$  and  $y_2$ , in that order, commanding the battalion unit currently at  $(x_1, y_1)$  to move to  $(x_2, y_2)$ .

## Sample Input

```
0
5 5
..##.
#...#
#000#
#.0#
.###.
```

## Sample Output

```
1
2 1 2 2
```

## Scoring

If your output is invalid (battalion units overlap, move out of bounds, or the final formation does not completely surround the research centers, etc.), then your score will be zero. Otherwise, for each test case, if the time determined by your program is  $ans$ , then your score out of 10 will be determined as follows:

$$\begin{cases} 10 & ans \leq A_i \\ 1 + \left\lfloor \left( \frac{ans - B_i}{A_i - B_i} \right)^2 \times 9 \right\rfloor & A_i < ans \leq B_i \\ 1 & B_i < ans \end{cases}$$

For each test case, there are two grading parameters  $A_i$  and  $B_i$  where  $A_i < B_i$ .

## Experimentation

We supply a tool [surround\\_check.py](#) for you to test whether your solutions are accepted. The usage for this program is:

```
surround_check.py <input-file> <output-file>
```

When you execute `surround_check.py`, the program will process your supplied input and output files and print the results to standard output. Possible results of the execution include:

- `Abnormal termination`: An unknown error occurred.
- `overlap`: Either an overlap between two battalion units occurred, or a battalion unit overlapped a research center in the final arrangement.
- `outside`: A battalion unit moved outside of the boundaries of the matrix.
- `move error`: Either an attempt was made to move a nonexistent battalion unit, or the distance of a move was not equal to a single unit.
- `not surround`: Battalions did not completely surround all research centers in the final arrangement.
- `time not match`: The number of lines in the output file did not correspond to the outputted answer.
- `yes`: The solution is valid; submit your solution to see its score.

Problem translated to English by [Alex](#).