CENG 202 Data Structures

Spring 2015

HOMEWORK-2

Due: 13 April 2015, 23:59

RULES

1. Late submission is not allowed.

2. **There is no teaming up.** The homework has to be done individually. In case of cheating, **all involved will get zero. You cannot make any objections to a grade 0. Your emails will not be responded in this case and your grade will never change for both sides**. Be very careful and honest about this rule!

3. Be careful about input/output specifications (do not print any unnecessary characters, whitespaces.) Your homeworks will be read automatically.

4. Codes not compiled will directly get 0.

5. Submit your homework electronically through MOODLE. No excuse for late submission even for 1 minute. In such a case, dont try to send emails to teacher/asistants. Emails will not be accepted and responded.

A CAMERA SYSTEM FOR A CLOSED PARKING LOT

In this homework, you are required to place cameras appropriately in a parking lot so that every part of the park can be watched by the owner of the lot. You will be given the map of the park. The camera system you will place is composed of cameras each of which can **view** the direct axis of four directions, **north, south, east and west**. The parking area is divided into unit squares. Each square is either empty or has a barrier which blocks the vision of that square and its behind. A square having the camera is assumed to be viewed by that camera.

An example view of a camera is shown in Figure 1. Here, the park is 5x4 grid. A camera is placed at square (1,1). There is a block at square (3,1).Yellow squares are the squares that are in the view of the camera placed at (1,1).

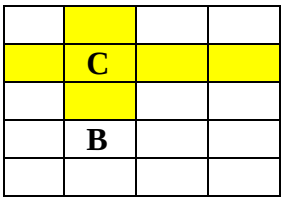

Figure 1: Example View of a Camera

Your program will setup a camera system for a given area where all of the empty squares of that area can be viewed by using this system. **In addition, the cameras you placed should not be in the view of each other in your solution. In other words, cameras can see only empty squares and blocks, they should be placed in such a way that they should not see each other.**

The size of the area, the locations of the blocks and the number of the cameras that will be placed will be given to you as an input file. An area of size **nxm that has b blocks and that will be controlled by c cameras** will be given to you in the following format:

input file:

```
n m
c
b
x1 y1
x2 y2
x3 y3
.
.
.
.
xb yb
```

The first line of the input file represents the dimensions of the area, the second line represents the number of cameras you will place, and the third line shows the number of the squares having the blocks. After the third line, the locations of the squares that has blocks are listed as x-y pairs. Each pair is located in one line seperated by one whitespace.

The top-left square of the area is represented as the square (0, 0) while the bottom-right square of the area is represented as (n-1, m-1) in an area of size nxm.

Your program should generate an output file that has the locations of the cameras. The locations of the c cameras should be printed in the output file as follows:

output file:

```
x1 y1
x2 y2
x3 y3
.
.
.
.
xc yc
```

An example of the execution of your program is given below. Assume that the parking lot is in a configuration as shown in Figure 2.

y / x	0	1	2	3	4
0	**B**				
1			**B**		**B**
2		**B**	**B**		
3			**B**		**B**

Figure 2: Example Parking Lot Configuration

Your input file for this lot is given below:

input file:

```
4 5
3
7
0 0
1 2
1 4
2 1
2 2
3 2
3 4
```

A solution of this example is shown below.

y / x	0	1	2	3	4
0	**B**	**C**			
1			**B**		**B**
2		**B**	**B**	**C**	
3	**C**		**B**		**B**

The output file should be:

output file:

```
0 1
2 3
3 0
```

Specifications:

1. The area can be at most 10x10.
2. Your class with the main function should be named **hw2.java.** You can implement any number of classes in your solution, however, the main function should be placed in **hw2.java.**
3. **hw2.java** class should take two command-line arguments, name of the input file and name of the output file.

4. Be careful about input/output specifications (do not print any unnecessary characters, white spaces) since black box testing will be used in evaluation, codes will not be opened.
5. Good Luck!