Middle East Technical University — CENG 140

Department of Computer Engineering — Spring 2015

Take home Exam 1

REGULATIONS

Due date: 17 April 2015, Friday, 23:59:59

Submission: Electronically. You will be submitting your program source code written in a file through the COW web system (see the specifications below for naming your file(s)). Resubmission is allowed (till the last moment of the due date), the last will replace the previous.

Team: There is **no** teaming up. The take home exam has to be done/turned in individually.

Cheating: This is an exam: all parts involved (source(s) and receiver(s)) get zero+parts will be subject to disciplinary action.

PROBLEM

Let's assume that a group of football fans visit İstanbul for a football tournament where 8 teams will participate. Each team has a name, but for brevitiy, we call them with the first letters of the alphabet, i.e., {A, B, C, D, E, F, G, H}. A camp is created out of the city in the form of an X*Y grid so that each cell of a grid is a tent where a fan stays. Each fan supports only one team.

Before the night the tournament starts, a particular fan, Bob, staying in the tent position *P(x, y)* (such that $0<=x<X$ and $0<=y<Y$) just got upset with the team he has already supported (let's call it the original team, **O**) and decides to support another team (let's call it the new team, **N**). However, as we all know, "one rotten egg spoils the whole basket"; so Bob visits all the neighbor tents in the camp to convince them to change their teams to support his new team **N**, as well. The neighbor tents of a tent at position *P* is all the tents that are **1-cell distance** to cell *P* in all possible directions (east, west, south, north, southeast, southwest, northeast, northwest). However, Bob will convince a neighbor to support team **N** *only if* the neighbor is also a supporter of the original team **O** that Bob supported before, and in this case, the neighbor is converted to support **N**. Each such converted fan (i.e., those who changed their teams from **O** to **N**) also starts visiting their own neighbor tents to repeat the same thing; that is, to convince their neighbors to support team **N** instead of team **O**. This process continues until no neighbor supporting the team **O** remains.

In this THE, you will be given *X*, *Y*, and the team of each fan in the camp at the beginning (as a matrix), as well as a tent position *P* of Bob and his new team, *N*. Starting from this initial tent *P*, you will convert the neighbors to team *N* as described above. At the end, you will print each row of the new matrix; i.e., the team of the fan in each tent in the camp.

SPECIFICATIONS

1. Assume *X* and *Y* can be at most 1000.

2. You are **not** allowed to use any dynamic data structures (as we have not discussed them in the class yet), so use multi-dimensional arrays.

3. You will read the input from standard input and write the resulting matrix to standard output. You may use re-direction for simplicity (see the example below and recitation notes).

4. You will solve this question in **two ways** and upload **two files** to COW:

a) Recursive solution: Your solution file must be uploaded as STUDENTID_recursive.c

b) Iterative solution: Your solution file must be uploaded as STUDENTID_iterative.c

5. Time bound: We will provide you some example input files with different sizes. Even for the largest files (for *X*=*Y*=1000), we observed that both recursive and iterative solutions work on `inek` machines in less than a minute. So, before submitting, be sure that your code works under a minute on the `inek` machines, as during grading it will be automatically killed after 60 seconds of execution.

6. *Hint:* If you develop your code somewhere other than `inek` machines, first try with the smaller inputs. If they run on the smaller inputs but fail for the larger ones; you may like to try them on `inek` machines, as the problem simply can be the physical capacity of your own computer.

Example:

The following is an example input file (let's say **inp1.txt**), where the first line has two integers, *X* and *Y*. Next, there are *X* lines and each has length *Y* (you may read each line as a string using the standard scanf() function). In the last line, the first two inputs are the *x* and *y* coordinates for the initial tent that is converted, and the following character is the new team. (Note that the table-style lines are only for illustration here, they do not appear in the actual file).

inp1.txt:

8 8

G	A	A	B	B	C	A	A
B	E	E	F	F	B	B	B
F	H	E	F	F	C	A	D
G	H	E	G	**F**	D	E	D
B	H	C	A	F	F	F	A
C	H	H	A	F	F	F	F
D	C	D	G	F	F	F	C
B	B	H	G	A	A	B	B

3 4 D

After Bob (shown as bold in the input) decides to support team D (instead of F), among his eight neighbors (shown as shaded below), only four will change their team to D, as they were

also supporting F (Bob's original team); whereas no team change happens for all the other neighbors.

G	A	A	B	B	C	A	A
B	E	E	F	F	B	B	B
F	H	E	F	F	C	A	D
G	H	E	G	**D**	D	E	D
B	H	C	A	F	F	F	A
C	H	H	A	F	F	F	F
D	C	D	G	F	F	F	C
B	B	H	G	A	A	B	B

This procedure is repeated by the "converted" neighbors of Bob's neighbors, and so on. At the end, the output will be as follows (tents that now support team D instead of F is shown as shaded):

out1.txt

G	A	A	B	B	C	A	A
B	E	E	**D**	**D**	B	B	B
F	H	E	**D**	**D**	C	A	D
G	H	E	G	**D**	D	E	D
B	H	C	A	**D**	**D**	**D**	A
C	H	H	A	**D**	**D**	**D**	**D**
D	C	D	G	**D**	**D**	**D**	C
B	B	H	G	A	A	B	B

If your executable is called as the1, you can call it with re-directions for input and output files as follows:

./the1 <inp1.txt >out1.txt