COMP 201 - Spring 2021
Assignment 3

Assigned: 25 March 2021 23:59, Due: 14 April 2021 23:59

Farzin Negahbani (fnegahbani19@ku.edu.tr) is the lead person for this assignment.

Introduction

The Covid-19 virus is dominating every single one of our lives at the moment and needs no introduction. New terms such as social-distancing are now part of our culture and everyday routines. Current recommendations are that people not get within six feet of each other. Clearly, this is a sudden and drastic change for every human on the planet. In this assignment, you will simulate and conclude why social distancing is critical and gain experience in using dynamic arrays and heap management. In the end, you can play with the parameters, especially the social distancing, and see the effect of complying with the social distancing rules.

Logistics

This is an individual assignment, and all hand-ins are electronic. Any clarification or correction will be announced on the Blackboard.

Handout Instructions

How to start

I Accept the GitHub Classroom assignment using the link: https://classroom.github.com/a/sgGyi8Ll

II Clone the GitHub repository created for you to a Linux machine in which you plan to do your work (Using Linux servers [linuxpool.ku.edu.tr]. See Section 4 for details is suggested.):

```
$ git clone https://github.com/COMP201-Spring2021/assignment-3-USER.git
```

(Replace USER with your GitHub username that you use to accept the assignment)

Task

You should implement a simulator that investigates the effect of social distancing in a grid world based on some input criteria. In the assignment template file, a base code that accepts some command-line arguments is provided. After taking input variables, your code should dynamically build a grid world that each cell that corresponds to a different person. Then, your code simulates a disease infection in a cyclic manner. In each cycle, your code evaluates and updates the latest situation of each person. Below, we described the specifications of the simulation and input parameters. You should keep in mind that, since you may need to keep track of each person's detail (e.g., infected, healthy or dead, if infected, the cycle that person is infected or etc.), you can define a person's information as a structures in c language that results in an array of structures for the grid world.

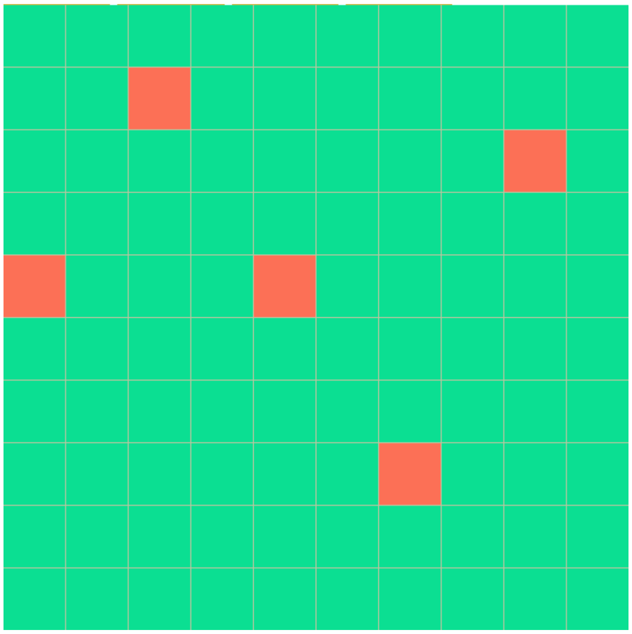


Figure 1: Sample of a 10 × 10 grid world in the initial with 5% active cases showed in red.

After the forth cycle, you may see a grid world similar to the following picture where you have both active cases and dead cases:

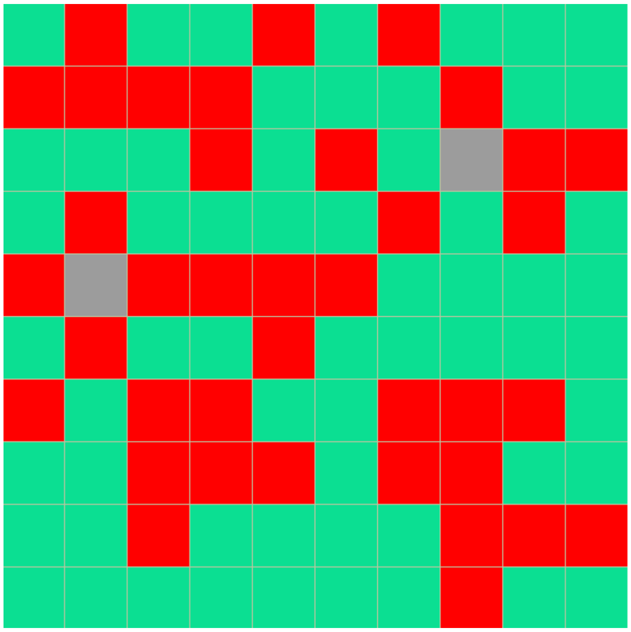


Figure 2: Sample of a 10 × 10 grid world after 4 cycles. Gray squares shows dead cases and red ones, represents the active cases.

Fixed Parameters

I **Grid World** assume a Grid World with size $N \times N$, which each cell corresponds to a person. Further, we assume, each person (cell) can transmit disease to only the neighbor cells.

II **Fatality Rate** should be set to 3%, and it means that 3% of the infected cases die after 4 cycles of getting infected (they are able to spread the disease for 4 cycles).

III **Recover Rate** the rest of 97% of infected people recover from the infection after 4 cycles, and they become immune for the rest of the simulation.

IV **Initial Active Cases** assumed 5% of the population as initial active cases. The population is defined by the grid world size and is equal to N^2 where N denotes the grid world size (check Input Parameters section for more details about the grid world size N).

Input Parameters

Based on recent studies, social distancing and suitable ventilation can be leveraged to decrease the infection rate. For simplicity, we assume all other parameters other than complying with the social distancing rules are fixed.

I **World Size** is the first parameter that specifies the size of your grid world that is denoted with N.

II **Social Distancing** shows the percentage of people who comply with the social distancing rules.

  I For example, 60% rate means in case of contact with infected people, there is 60% chance of transmitting the disease.

III **Simulation length** defines how many cycles your code should continue the simulation. In each step, you evaluate the status of each person and update your world.

IV **Random Seed** To be able to evaluate your code, you should provide a random seed for all of your random functions that can be passed using the last command-line argument.

  I For more details on random and random seed please check this link.

The way we expect your code should be called in a case where world size is 10, social distancing compliance is 75%, Simulation length is 100, and the random seed is 7.

```
./main 10 75 100 7
```

Return Values

This section describes the return information and files of the simulation. Your code should print the following information in the end of the simulation.

I **Total Infections** during the simulation.

II **Total death count** during the simulation.

III **Total Recovered cases** during the simulation.

IV **Max active cases** peak of active cases, during the simulation.

V **Cyclic visualization** Your code should provide a file for each cycle (CYCLENUM.txt) that shows that cycle's status of the grid world. For simplicity, you should use ASCII art where '0', '1', and '*' denotes dead person, healthy person, and active case and characters are separated by one space. You can check the example below.

```
1 1 1 1 1 1 1 1 1 1
1 * 1 1 1 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 * * 1 1 1 1 1
1 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 * 1 1 1 1 1 * 1 1
1 1 1 1 * 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

1 Evaluation

Your score will be computed out of a maximum of 100 points based on the following distribution:

65 Correctness points.

15 Visualization points.

10 Effective use of version control points.

10 Style points.

***20** Bonus points for any creative additions or better visualizations.

Correctness points. Your code will be evaluated based on a set of given input parameters and should result in correct numbers for 1. Total Infections 2. Total death count 3. Total Recovered cases and 4. Max active cases.

Visualization points. Your code should generate *.txt files that demonstrate each cycle's status using ASCII art.

Effective use of version control points. You are required to push your changes to the repository frequently. If you only push the final version, even if it is implemented 100% correctly, you will lose a fraction of the grade because you are expected to learn to use Version Control Systems effectively. You do not have to push every small piece of change to GitHub, but every meaningful change should be pushed. For example, each of the functions coded and tested can be one commit. **For each function, there should be at least one commit (with proper commit message) that includes just modifications on that function.**

Style points. Finally, we've reserved 10 points for a subjective evaluation of the style of your solutions and your commenting. Your solutions should be as clean and straightforward as possible. Your comments should be informative, but they need not be extensive.

Important Note: We use automated plagiarism detection to compare your assignment submission with others and also the code repositories on GitHub and similar sites. Moreover, we plan to ask randomly selected 10% of students to explain their code verbally after the assignments are graded. And one may lose full credit if he or she fails from this oral part.

2 Handin Instructions

Same as previous assignments, we use GitHub for the submissions as follows. Note that we want you to get used to using a version management system (Git) in terms of writing good commit messages and frequently committing your work so that you can get the most out of Git.

I Commit all the changes you make: `$ git commit -a -m "commit message"`
Note: please use meaningful commit messages because

II Push your work to GitHub servers: `$ git push origin main`

3 Advice

- Keep your GitHub repository by frequently committing the changes.
- Don't forget GDB and Valgrind since they can save a lot of time in debugging.
- Use linuxpool.ku.edu.tr Linux servers to test your code in order to avoid compatibility issues.
- Since you will learn how to write a makefile in lab 5, you should write a makefile for your own project that helps you compile and run your code faster.

4 How to use linuxpool.ku.edu.tr linux servers

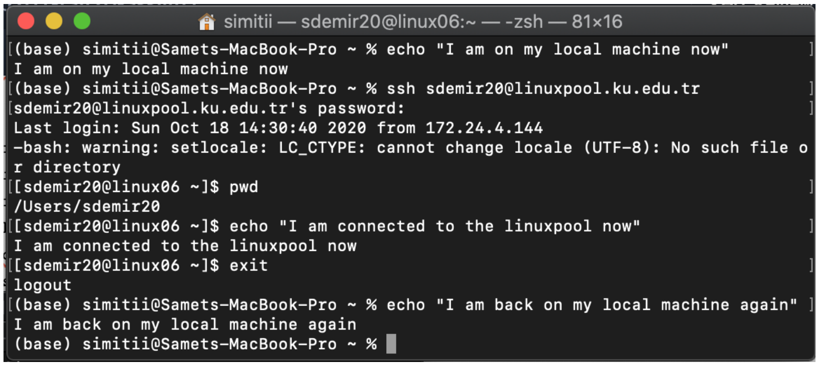
I Connect to KU VPN (If you are connected to the KU network, you can skip this step.)
See for details: https://confluence.ku.edu.tr/kuhelp/ithelp/it-services/network-and-wireless/vpn-access

II Connect to linuxpool.ku.edu.tr server using SSH (Replace USER with your Koç University username):
`$ ssh USER@linuxpool.ku.edu.tr`
(It will ask your password, type your Koç University password.)

III When you are finished with your work, you can disconnect by typing: `$ exit`

Your connection to the server may drop sometimes. In that case, you need to reconnect.

We advise you to watch the following video about the usage of SSH, which is used to connect remote servers, and SCP, which is used to transfer files between remote servers and your local machine: https://www.youtube.com/watch?v=rm6pewTcSro



```
[(base) simitii@Samets-MacBook-Pro ~ % echo "I am on my local machine now"
 I am on my local machine now
[(base) simitii@Samets-MacBook-Pro ~ % ssh sdemir20@linuxpool.ku.edu.tr
[sdemir20@linuxpool.ku.edu.tr's password:
 Last login: Sun Oct 18 14:30:40 2020 from 172.24.4.144
 -bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file o
 r directory
[[sdemir20@linux06 ~]$ pwd
 /Users/sdemir20
[[sdemir20@linux06 ~]$ echo "I am connected to the linuxpool now"
 I am connected to the linuxpool now
[[sdemir20@linux06 ~]$ exit
 logout
[(base) simitii@Samets-MacBook-Pro ~ % echo "I am back on my local machine again"
 I am back on my local machine again
 (base) simitii@Samets-MacBook-Pro ~ %
```

Figure 3: How to connect and disconnect using SSH

5 Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss the given assignments with your classmates, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else’s work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the web material as everything on the web has been written by someone else. See Koç University - Student Code of Conduct.

6 Late Submission Policy

You may use up to 7 grace days (in total) over the course of the semester for the assignments. That is, you can submit your solutions without any penalty if you have free grace days left. Any additional unapproved late submission will be punished (1 day late: 20% off, 2 days late: 40% off) and **no submission after 2 days will be accepted.**

Acknowledgement

This assignment is adapted from ”Steven Bitner, Stanford University” and LateX template from ”Samet Demir, Koc University”.