CSCE 1030: Homework 2

Due: 11:59 PM on Wednesday, September 27, 2017

PROGRAM DESCRIPTION:

The purpose of this programming project is to write a C++ program to simulate a disease outbreak and determine how many days the outbreak lasted using user input, branch statements, and loops.

BACKGROUND:

In disease modelling, one of the basic models is called the SIR model. SIR stands for Susceptible, Infectious, and Recovered, which represent the different states a person can be in with regard to an illness. A susceptible person is someone who is not sick, but can become sick. An infectious person is someone who is sick and can infect others with the disease. A recovered person is someone who is no longer sick and has developed an immunity to the disease.

To represent the process of being healthy, becoming sick, and recovering, we can use simple differential equations to compute the diseases progress over time. In this case we will be assuming each time step represents one day. To calculate the number of susceptible, infectious, and recovered individuals each day we can use the following equations:

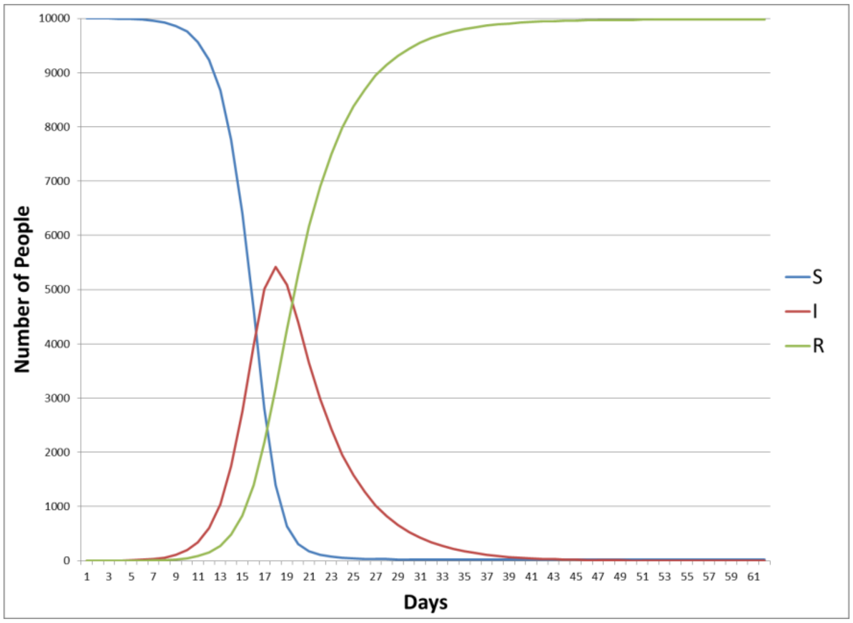
$$\boldsymbol{S_i = S_{i-1} - \beta I_{i-1} S_{i-1}}$$

$$\boldsymbol{I_i = I_{i-1} + \beta I_{i-1} S_{i-1} - \gamma I_{i-1}}$$

$$\boldsymbol{R_i = R_{i-1} + \gamma I_{i-1}}$$

In these equations, S_i is the number of susceptible people on day i, I_i is the number of infectious people on day i, R_i is the number of recovered people on day i, β is the contact rate (the rate at which people come into contact with infectious individuals and become infected), and γ is the recovery rate (the rate at which people recover from the disease). Additionally, since we are using differential equations, to calculate the values of S_i, I_i, and R_i, you need the values of the previous day's S_i, I_i, and R_i, values S_{i-1}, I_{i-1}, R_{i-1}.

A typical graph of the progression of an outbreak might look like the following, where the number of susceptible individuals decreases over time, the number of infectious individuals grows, peaks, and then decreases, and the number of recovered individuals grows over time.



REQUIREMENTS:

- As with all homework programs in this course, your program's output will initially display the department and course number, your name, your EUID, and your email address.
- You need to prompt the user for the total number of people in the simulation. This should be positive integers (i.e. > 0). If the user does not enter positive integers you should display a meaningful error message and continue to re-prompt the user to enter a positive integer.
- You will need to prompt the user for the initial number of infectious individuals. This should be positive integers (i.e. > 0) and should be less than the total number of people in the simulation. If the user does not enter an acceptable number you should display a meaningful error message and continue to re-prompt the user to enter a positive integer that is less than the total number of people in the simulation.
- You will need to obtain a value for β. This should be a small, positive number in the range of $0 < \beta < 1$. If the user does not enter an acceptable number you should display a meaningful error message and continue to re-prompt the user to a small, positive number in the range of $0 < \beta < 1$.
- You will need to obtain a value for γ. This should be a small, positive number in the range of $0 < \gamma < 1$. If the user does not enter an acceptable number you should display a meaningful error message and continue to re-prompt the user to a small, positive number in the range of $0 < \gamma < 1$.

- You must calculate the number of susceptible individuals by subtracting the infectious individual count from the total number of people in the simulation. The number of recovered individuals is assumed to be 0 at the beginning of the simulation.
- Because you are using differential equations in this simulation, will need to set S_{i-1}, I_{i-1}, R_{i-1} to the initial values of the number of susceptible individuals, infectious individuals, and recovered individuals, respectively.
- In order to determine how long the outbreak lasts, you will need to count the number of days until the number of infectious individuals is 0. You can simulate this using a while loop, where each iteration of the loop corresponds to one day, and a counter variable which keeps track of which day it is. Inside the loop, you will need perform the following calculations until the total number of infectious individuals is 0:
    - Update the total number of susceptible individuals using the appropriate equation and variables.
    - Update the total number of infectious individuals using the appropriate equation and variables.
    - Update the total number of recovered individuals using the appropriate equation and variables.
    - Output the updated total number of susceptible, infectious, and recovered individuals along with the current day number. You should consider using `printf()` to make your output more readable.
    - Update the values of the previous day's susceptible, infectious, and recovered individuals counts (S_{i-1}, I_{i-1}, R_{i-1}) to the current day's susceptible, infectious, and recovered individuals counts (S,I,R).
    - Increment the day counter by one.
- Once the total number of infectious individuals reaches 0, you should output to the user the total number of days the outbreak lasted. This output should be in the form of a meaningful message describing the outbreak.
- Your code should be well documented in terms of comments. For example, good comments in general consist of a header (with your name, course section, date, and brief description), comments for each variable and commented blocks of code.
- Your program source code should be named "**homework2.cpp**", without the quotes.
- Your program will be graded based largely on whether it works correctly on the CSE machines (e.g., cse01, cse02, …, cse06), so you should make sure that your program compiles and runs on a CSE machine.
- **This is an individual programming assignment that must be the sole work of the individual student.**

You may assume that all input will be of the appropriate data type, although the range (e.g., a positive integer) may not be valid. Please pay attention to the SAMPLE OUTPUT for specific details about the flow and input/output of the program.

You shall use techniques and concepts discussed in class – you are not to use global variables, goto statements, or other items specifically not recommended in this class.

DESIGN(ALGORITHM):

On a piece of paper (or word processor), write down the algorithm, or sequence of steps, that you will use to solve the problem. You may think of this as a "recipe" for someone else to follow. Continue to refine your "recipe" until it is clear and deterministically solves the problem. Be sure to include the steps for prompting for input, performing calculations, and displaying output. This will be particularly helpful on this homework, as you have multiple variables you are working with, so writing it down can help you keep them in the right order.

You should attempt to solve the problem by hand first (using a calculator as needed) to work out what the answer should be for a few sets of inputs.

Type these steps and calculations into a document (i.e., Word, text, or PDF) that will be submitted along with your source code. Note that if you do any work by hand, images (such as pictures) may be used, but they must be clear and easily readable. This document shall contain both the algorithm and any supporting hand-calculations you used in verifying your results.

SAMPLE OUTPUT (input shown in bold green):

```
jeh0289@cse04:~/csce1030/fa17$ ./a.out
**********************************************
     Computer Science and Engineering
      CSCE 1030 - Computer Science I
 Student Name    EUID   euid@my.unt.edu
**********************************************
Welcome to the outbreak simulator.
Please enter a positive, whole number for the total of the population you
wish to simulate: 1000
Please enter a positive, whole number that is less than the total population
for the number of infectious individuals: 1
Please enter a positive number greater than 0 and less than 1 for the contact
rate: 0.0014
Please enter a positive number greater than 0 and less than 1 for the
recovery rate: 0.2
Day:   0 S:   997 I:     2 R:     0
Day:   1 S:   994 I:     4 R:     0
Day:   2 S:   988 I:     8 R:     0
Day:   3 S:   976 I:    17 R:     1
```

```
Day:   4 S:   952 I:    36 R:     4
Day:   5 S:   904 I:    76 R:    11
Day:   6 S:   807 I:   156 R:    26
Day:   7 S:   630 I:   301 R:    57
Day:   8 S:   364 I:   506 R:   117
Day:   9 S:   106 I:   662 R:   218
Day:  10 S:     7 I:   627 R:   350
Day:  11 S:     0 I:   507 R:   475
Day:  12 S:     0 I:   405 R:   576
Day:  13 S:     0 I:   323 R:   657
Day:  14 S:     0 I:   258 R:   721
Day:  15 S:     0 I:   206 R:   772
Day:  16 S:     0 I:   164 R:   813
Day:  17 S:     0 I:   131 R:   845
Day:  18 S:     0 I:   104 R:   871
Day:  19 S:     0 I:    83 R:   891
Day:  20 S:     0 I:    66 R:   907
Day:  21 S:     0 I:    52 R:   920
Day:  22 S:     0 I:    41 R:   930
Day:  23 S:     0 I:    32 R:   938
Day:  24 S:     0 I:    25 R:   944
Day:  25 S:     0 I:    19 R:   949
Day:  26 S:     0 I:    15 R:   952
Day:  27 S:     0 I:    11 R:   955
Day:  28 S:     0 I:     8 R:   957
Day:  29 S:     0 I:     6 R:   958
Day:  30 S:     0 I:     4 R:   959
Day:  31 S:     0 I:     3 R:   959
Day:  32 S:     0 I:     2 R:   959
Day:  33 S:     0 I:     1 R:   959
Day:  34 S:     0 I:     0 R:   959
The outbreak took 35 days to end.

jeh0289@cse04:~/csce1030/fa17$ ./a.out
*****************************************
     Computer Science and Engineering
      CSCE 1030 - Computer Science I
 Student Name    EUID   euid@my.unt.edu
*****************************************
Welcome to the outbreak simulator.
Please enter a positive, whole number for the total of the population you
wish to simulate: 0
0 is not a positive whole number. Please enter a positive whole number for
the total population: 100
Please enter a positive, whole number that is less than the total population
for the number of infectious individuals: 101
101 is not a positive whole number or is greater than the total population
100. Please enter a positive whole number for the number of infected
individuals: 1
```

```
Please enter a positive number greater than 0 and less than 1 for the contact
rate: 2
2 is not greater than 0 and less than 1. Please enter a positive number
greater than 0 and less than 1 for the contact rate: .014
Please enter a positive number greater than 0 and less than 1 for the
recovery rate: 2
2 is not greater than 0 and less than 1. Please enter a positive number
greater than 0 and less than 1 for the recovery rate: .2
Day:   0 S:    97 I:     2 R:     0
Day:   1 S:    94 I:     4 R:     0
Day:   2 S:    88 I:     8 R:     0
Day:   3 S:    78 I:    16 R:     1
Day:   4 S:    60 I:    30 R:     4
Day:   5 S:    34 I:    49 R:    10
Day:   6 S:    10 I:    62 R:    19
Day:   7 S:     1 I:    58 R:    31
Day:   8 S:     0 I:    47 R:    42
Day:   9 S:     0 I:    37 R:    51
Day:  10 S:     0 I:    29 R:    58
Day:  11 S:     0 I:    23 R:    63
Day:  12 S:     0 I:    18 R:    67
Day:  13 S:     0 I:    14 R:    70
Day:  14 S:     0 I:    11 R:    72
Day:  15 S:     0 I:     8 R:    74
Day:  16 S:     0 I:     6 R:    75
Day:  17 S:     0 I:     4 R:    76
Day:  18 S:     0 I:     3 R:    76
Day:  19 S:     0 I:     2 R:    76
Day:  20 S:     0 I:     1 R:    76
Day:  21 S:     0 I:     0 R:    76
The outbreak took 22 days to end.
```

TESTING:

Test your program to check that it operates as desired with a variety of inputs. Then, compare the answers your code gives with the ones you get from hand calculations.

Perhaps try:

Total Population = 100, Number of infectious = 1, Contact Rate = 0.02, Recovery rate = 0.65

Total Population = 100, Number of infectious = 5, Contact Rate = 0.015, Recovery rate = 0.55

Total Population = 100, Number of infectious = 30, Contact Rate = 0.01, Recovery rate = 0.4

SUBMISSION:

Your program will be graded based largely upon whether it works correctly on the CSE machines, so you should make sure your program compiles and runs on the CSE machines.

Your program will also be graded based upon your program style. This means that you should use comments (as directed), meaningful variable names, and a consistent indentation style as recommended in the textbook and in class.

We will be using an electronic homework submission on Blackboard to make sure that all students hand their programming projects on time. You will submit both (1) the program source code file and (2) the algorithm design document to the Homework 2 dropbox on Blackboard by the due date and time.

Note that this project must be done individually. Program submissions will be checked using a code plagiarism tool against other solutions, so please ensure that all work submitted is your own.

Note that the dates on your electronic submission will be used to verify that you met the due date and time above. All homework up to 24 hours late will receive a 50% grade penalty. Later submissions will receive zero credit, so hand in your best effort on the due date.

As a safety precaution, do not edit your program (using vi or pico) after you have submitted your program where you might accidentally re-save the program, causing the timestamp on your file to be later than the due date. If you want to look (or work on it) after submitting, make a copy of your submission and work off of that copy. Should there be any issues with your submission, this timestamp on your code on the CSE machines will be used to validate when the program was completed.