

Assessing Infection

Prerequisites, Goals, and Outcomes

Prerequisites: Students should have mastered the following prerequisite skills.

- *Recursion* - Understanding of the concept of a recursive function
- *Divide and Conquer* - Understanding of the divide and conquer approach to solving a problem
- *Vectors* - Use of the STL vector container

Goals: This assignment is designed to reinforce the student's understanding of recursion and problem solving using a divide and conquer approach.

Outcomes: Students successfully completing this assignment would master the following outcomes.

- Understand recursion
- Understand and apply a divide and conquer approach to solving a problem

Background

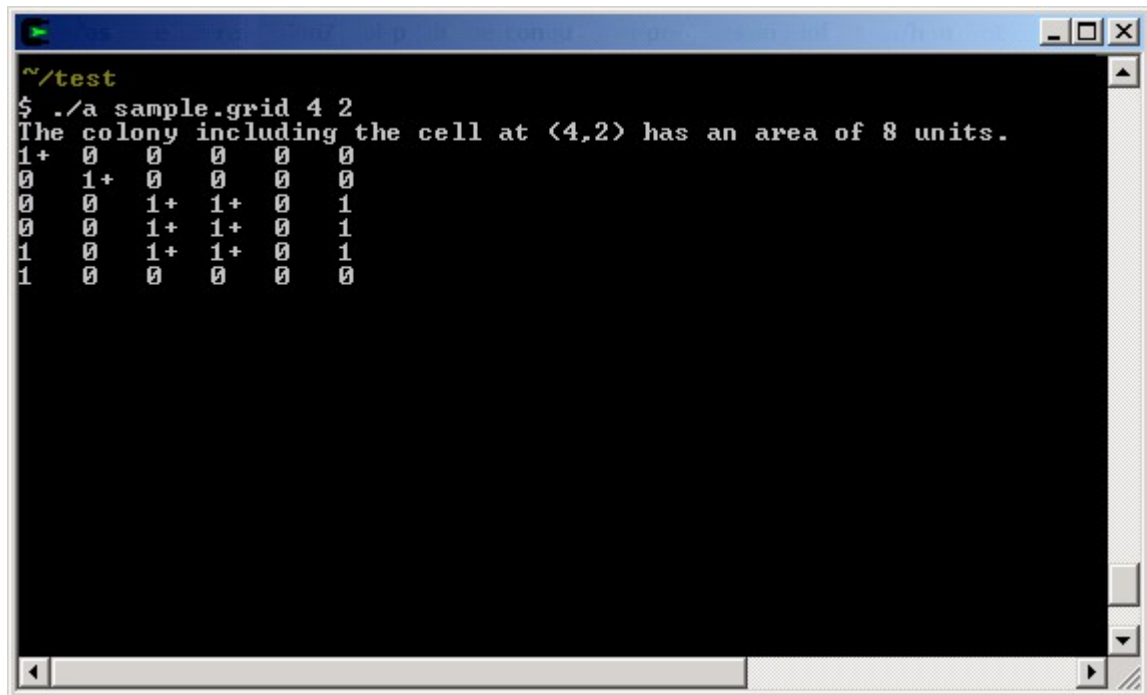
According to the World Health Organization, infectious disease ranks as the leading cause of death in the world. In 1998 alone, over 17 million people died from infectious and parasitic diseases such as acute lower respiratory infections, tuberculosis, HIV/AIDS, and malaria. It is forecast that infectious disease will continue to kill millions of people, especially those living in developing countries.

The medical profession and scientific community of the world are fighting the infectious disease threat with new tools and technologies from a variety of fields. From this effort, a new field of research has emerged. *Infectious Disease Epidemiology* is the study of the variables that influence the growth and spread of infectious diseases. This relatively new field combines molecular biology, immunology, genetics, and the computational sciences. A focus of this field is the study of the factors that influence the growth of an infectious disease within a single organism, and the factors that influence the pattern of infection across an entire population.

Description

This assignment asks you to finish the implementation of a program that assesses the level of infection in a tissue sample. You are given data representing a rectangular tissue sample, overlaid with a grid. Certain portions of the tissue are infected; others are not. Your goal is to help assess the extent of the infection by writing a program that, given the coordinates of a colony of infection, can determine its size.

A typical use of the program follows. The user interacts with the program only through command-line arguments. The user supplies to the program a data filename and the coordinates of a cell in the grid. The coordinates are specified by row and then column, both starting at zero. The program calculates the extent of infection at that coordinate and outputs a two-dimensional representation of the tissue sample. Figure 1 depicts the execution of the program.



```
~/test
$ ./a sample.grid 4 2
The colony including the cell at <4,2> has an area of 8 units.
1+  0  0  0  0  0
0   1+ 0  0  0  0
0   0  1+ 1+ 0  1
0   0  1+ 1+ 0  1
1   0  1+ 1+ 0  1
1   0  0  0  0  0
```

Figure 1 Output from a sample solution

For the purpose of this assessment, we consider a "colony" of infected tissue to be a set of adjacent and infected cells. In Figure 1, we can see three separate colonies. The smallest colony consists of two cells and is located in the lower left corner of the grid. Another colony consisting of three infected cells exists on the far right edge of the grid. The largest colony of eight cells resides primarily in the middle of the grid. This colony has a small arm into the upper left corner of the grid. Notice from this colony that cells residing in diagonals are considered "adjacent." The plus signs next to the cells in this largest colony indicate that they all belong to the colony that contains the user entered coordinate.

Files

Following is a list of files needed to complete this assessment.

- [me.arj](#) contains all of the following necessary files:
 - *main.cpp* - This file contains the main routine. You do not need to modify this file.
 - *grid.h* - This file contains the declaration of class *grid*.
 - *grid.cpp* - This file contains the implementation of class *grid*.
 - *sample.grid* - A data file containing sample tissue data. This file was used during the creation of the screen shot in Figure 1.

Tasks

To complete this assessment, you need to complete the implementation of class *grid*.

To begin, verify the files needed for this assessment.

1. **Extract** the archive to retrieve the files needed to complete this assessment.

Following is an ordered list of steps that serves as a guide to completing this assessment. Work and test incrementally. Save often

1. **Begin** by studying the program and class *grid* in its original state. Compile and run the program to get a feel for the command-line interface. Use the *sample.grid* data file. Make sure you understand how the main routine uses class *grid*, and what the class *grid* constructor accomplishes. Pay particular attention to how class *grid* represents the cell data. Even though the tissue sample is a two-dimensional object, a one-dimensional data structure (a vector) is used for representation.
2. **Finish** the implementation of function *grid::count*. This routine should use a divide-and-conquer recursive approach to solve the problem. This means you will have two or more recursive calls in function *grid::count*. To correctly complete function *grid::count*, you will have to add additional data members and/or functions to class *grid*. You will also have to alter other existing code. For instance, you will have to alter the overloaded *<<* operator implementation to output a plus sign (+) next to cells that belong to a colony that subsumes the user entered coordinate.

Submission

Submit **only** the following.

1. *grid.h* - your updated class *grid* declaration
2. *grid.cpp* - your updated class *grid* implementation