CS 2123 Data Structures
Recitation 3
Due Friday February 26

Difficulty ⋆⋆ (out of 5)

1. (100 pts) Write a program to read a two dimensional array from a file and compute the sum of the elements in a rectangular area specified by the user efficiently in $O(1)$ time without using loops.

   Input file format is as follows.

   ```
   norows nocols
   row0
   row1
   ...
   row norows-1
   ```

   Consider the following input file *a.txt* as an example

   ```
   4 5
   1 1 1 1 1
   1 1 1 1 1
   1 1 1 1 1
   1 1 1 1 1
   ```

   In this representation, each element represents a single value and $A[0][0]$ is the top left entry in the matrix. To compute the sum of elements in a rectangular area in $O(1)$ time, use a new representation where an element $B[i][j]$ denotes the sum of all the elements $A[x][y]$ where $0 \leq x \leq i$ and $0 \leq y \leq j$. Convert the array read from the file into this representation. With this new representation the value of the array elements for the above array are as follows.

   ```
   1 2  3  4  5
   2 4  6  8 10
   3 6  9 12 15
   4 8 12 16 20
   ```

   Using this new representation the sum of a rectangular area can be computed without using a loop. Consider the diagram in Figure 1. We are interested in Area A_4 and this can be computed using A_1, A_2, A_3, A_4. Try to find the expression to find the sum of elements in area A_4. Note that each entry $B[i][j]$ stores the sum of all the elements $A[x][y]$ where $0 \leq x \leq i$ and $0 \leq y \leq j$.

   Get the coordinates of a rectangular area from the user and compute the sum of the elements in that area using the above technique.

   Sample execution for the file *a.txt* is given below

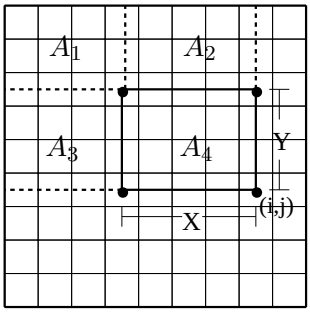

Figure 1: Finding the sum of numbers in a rectangular area

```
Enter top left and bottom right coordinates of rectangle (-1 to quit)
0 1 0 3
Sum = 3
0 0 2 3
Sum = 12
1 1 3 3
Sum = 9
```

Dynamically allocate the array after reading the dimensions from the file. Free it before exit and use *valgrind* for potential memory errors.

Test your program with different files. Make sure your program works with special cases. You can create your own sample files.

Submit your program electronically using the blackboard system. Only one member of the group should send it. List the name of the group members on the top of your submission.