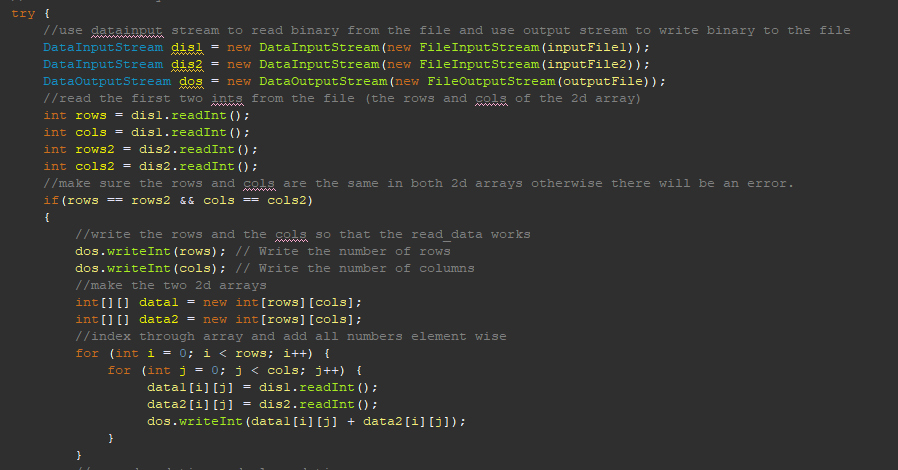
William Morgan   
  
Homework 3 write up

General overview of all programs:

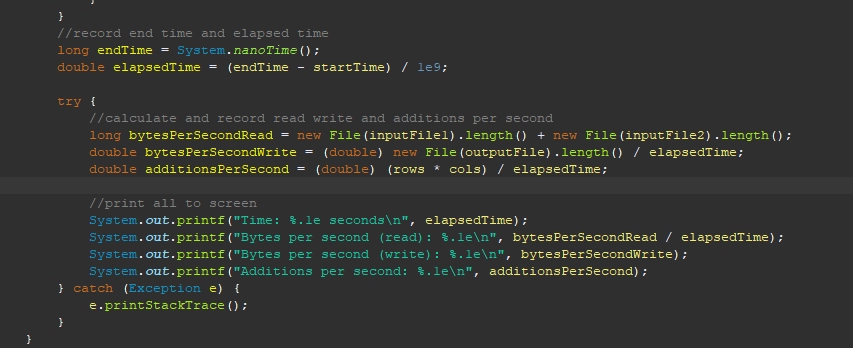
All of the programs I created use DataInputStream or DataOutputStream. These are imported classes that write and read binary files. Using a scanner would not work in this situation because we need to write and read binary files. These specialized classes are seen throughout my programs. Every time one of the objects is mentioned it is placed inside of a try catch just in case the file is not found. When creating one of those objects you must do something like this “new DataInputStream(new FileInputStream(inputFile1))” because the object requires a file input/output stream for the constructor. The data output stream works by converting the data into bytes which is then written into a file. The data input stream converts the bytes into data types which are then read by the program. Each program also records the time that it takes to run. Of course, at the start of the program it gets the time in nanoseconds and end time in nanoseconds then calculates the total run time of the program. Each program prints the runtimes in a format of scientific notation to one decimal place. Each program also checks for the number of arguments in the args array so that we know how many arguments are needed and the user knows what to enter. If the argument check is pased the program, then moves on to read the data from the args array. I have included this general overview so that I do not have to say the same thing over again.

Add\_data\_2d:

This program starts inside of a try catch loop and then creates the DataInputStream and DataOutputStream objects (explanation of objects shown above). After that we collect the number of rows and columns from each file. Once that is complete, we check to make sure that the rows and columns are the same from each file. We do this to avoid any array of out of bounds exceptions. If that criterion is passed, we write the rows and columns to the file for the next program to read. Then the 2d arrays are created. The next step is iterating through the 2d arrays using a nested for loop. This loop reads the ints from the files and then writes the addition of the ints to the new file.



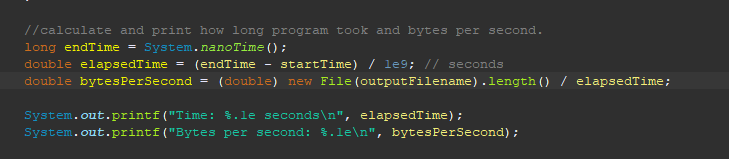
Once that is completed, we calculate the total elapsed time of the project. The next step is calculating the bytes per second read and written and the additions per second. We do this by finding the length of the input files (for the bytes read per second) and finding the length of the output file (for the bytes written per second). Also, the additions per second is calculated by multiplying the rows and columns together. Everything I just said about the additions, read, and write is then divided by the total elapsed time of the project to get the per second amounts.



Make\_data\_2d:

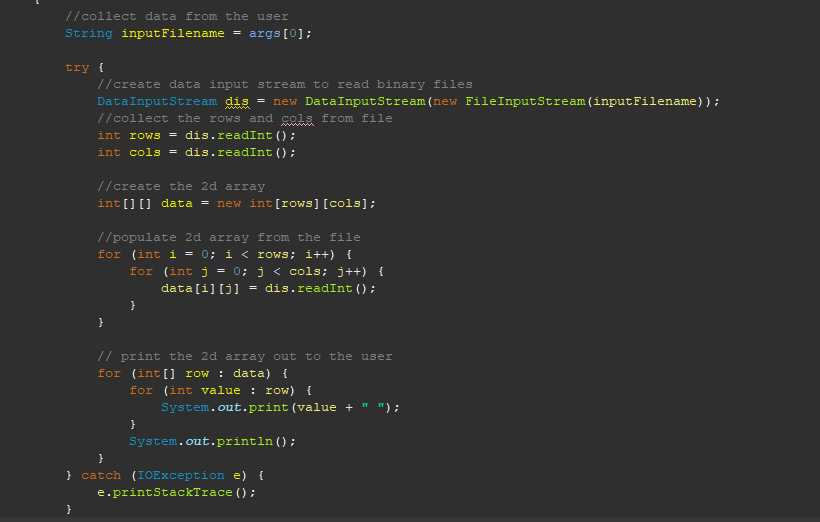
This program starts off by creating the 2d array from the user input. Then the random object is created so that we can create a random number with the given criteria. The next thing that executes is the nested for loop that populates the 2d array with random integers between the given bounds. This for loop is controlled by the rows and cols given so that there will never be an out of bound exception. The next thing that happens is creating the DataOutputStream to write binary into our file. After that we write the rows and columns to the file for convenience for the other programs. Once completed, I create another nested for loop controlled by the rows and columns to write the array data into the file. 

After that we use the same logic as the add\_data\_2d to calculate the total elapsed time and bytes per second.

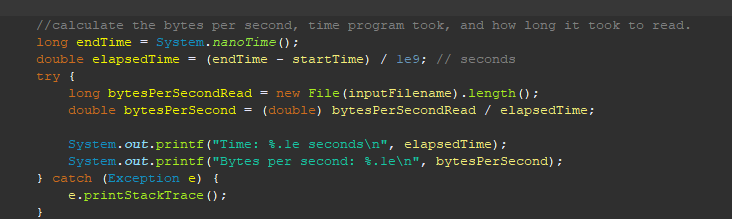


Read\_data\_2d:

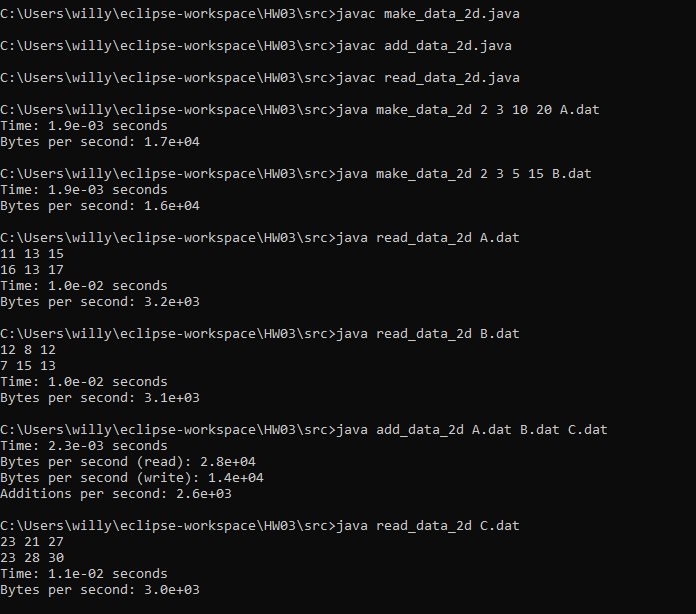
This program takes the data given by the user and then creates a DataInputStream to read the binary file. Next, we read the rows and columns from the file because we wrote it in our other programs for ease of use. Next the 2d array is created. We then create a new array and iterate through it using a nested for loop. We populate the array inside of this nested loop and this is done with the DataInputStream object. After that we print the data to the screen using another nested for loop. This loop has an initial logic of “for(int[] row: data)” this is saying that we will go through the data array one row at a time (for each row in data). Then we use “for(int value : row)” this is saying that we will take the value from each row/the row array created in the initial loop (for each value in row). Then it prints to the screen

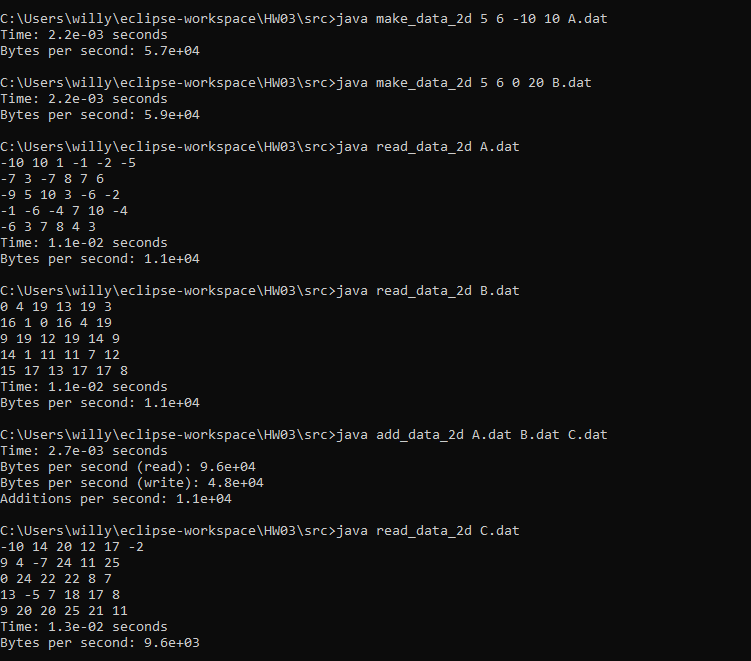


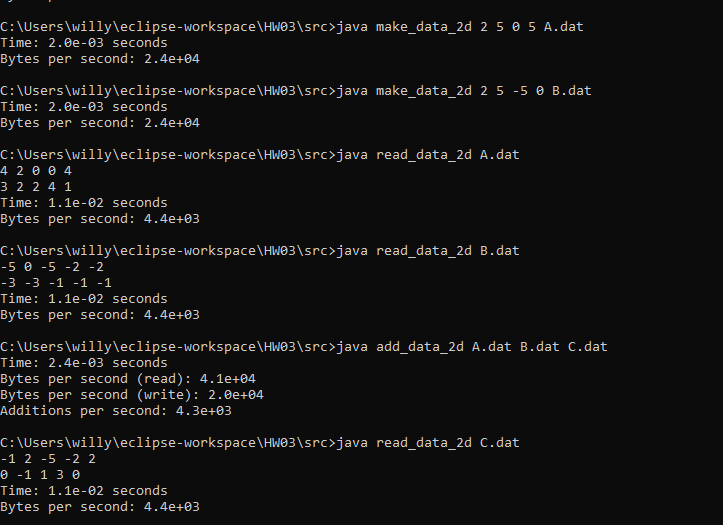
After that the bytes per second and total time the program took is printed to the screen using the same logic as before.

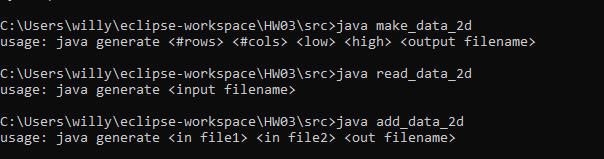


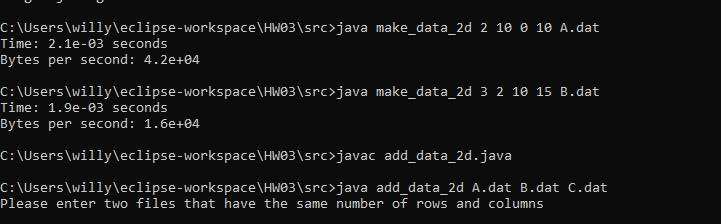
Homework 3 screen shots











File inside of zip

