

```
import java.net.*;
import java.io.*;
import java.util.*;

class readThread extends Thread{

// Implement the read thread features as explained in the above project description
// Look at the programming manual for more samples of read threads
// You can refer to the number of processes as multicastSenderReceiver.numProcesses
// numProcesses is declared as a static variable in the multicastSenderReceiver class

}

class multicastSenderReceiver{

    static int numbers[];
    static int numProcesses;

    public static void main(String[] args){

        try{

            numProcesses = Integer.parseInt(args[0]);

            int processID = Integer.parseInt(args[1]);

            int arraySize = Integer.parseInt(args[2]);

            numbers = new int[arraySize];

            String filename = args[3];

            FileReader fr = new FileReader(filename);
            BufferedReader br = new BufferedReader(fr);

            for (int i = 0; i < arraySize; i++){

                numbers[i] = Integer.parseInt(br.readLine() );

            }

            // Compute the local partial sum value and print it

            int multicastPort = 3456;
                // change it to the last 4 digits of your J# or 10000 plus your last 4 digits of J #

            // Complete the rest of the code to launch the read thread,
            // wait for the read threads of all the processes to start, before sending the partial
            sum value
            // send <key, value> as a string where key is the process ID and value is the partial
            sum value

        }
        catch(Exception e){e.printStackTrace();}

    }

}
```

}
}