```java
/*-----------------------------------------------
------Hussain Ahmad--------SP22_BCS_067--------
-----------------------------------------------
------Assignment 2---------Exercise 7.36--------
---------------------------------------------*/
import java.util.Scanner;
public class SP22_BCS_067
{
    public static void main(String[] args)
    {
        final int READ = 10;
        final int WRITE = 11;
        final int LOAD = 20;
        final int STORE = 21;
        final int ADD = 30;
        final int SUBTRACT = 31;
        final int DIVIDE = 32;
        final int MULTIPLY = 33;
        final int BRANCH = 40;
        final int BRANCHNEG = 41;
        final int BRANCHZERO = 42;
        final int HALT = 43;
        final int SENTINEL_NEG = 51; // 7.36(a) Sentinel loop to read 10 +ve numbers, display sum
        final int COUNTER_7 = 52; // 7.36(b) Counter-controlled loop to read 7 numbers, display
        average
        final int COMPARE = 53; // 7.36 (c) Read set of numbers and comapre for greatest, first
        num decides the set of reads

        Scanner input = new Scanner(System.in);

        int accumulator = 0;
        //creating Simpletron memory as an array of 100 integer elements
        int array[] = new int[100];

        boolean found = false;
        for(int i = 0; i < args.length; i++){
            if(args[i].length() != 4)
                found = true;
        }
        //if no arguments passed or argument length not 4, the program will end
        if(args.length == 0 | found == true){

            if(args.length == 0)
                System.out.printf("%s%n", "No arguments passed!");

            if(found == true)
                System.out.printf("%s%n", "Inavlid argument");
        }
        else
        {
            //populating the new array with arguments passed at the start
            for(int i = 0; i < args.length; i++){
                array[i] = Integer.parseInt(args[i]);
                //System.out.printf("index %s = %s%n", i,  args[i]);
            }
            System.out.println();
            int i = 0;
            int opcode = 0;
            int operand = 0;
            int instructionReg = 0;

            System.out.printf("%s%n%n", "====================================");
            for(i = 0; i < args.length; i++){

                instructionReg = (Integer.parseInt(args[i]));
                opcode = instructionReg / 100; //operation code
                operand = instructionReg % 100; //operand code

                switch(opcode)
                {
                    // input/output operations
                    case READ: //Read a word from the keyboard into a specific location in memory
                        System.out.printf("READ at index %02d : ", operand);
                        array[operand] = input.nextInt();
                        break;

                    case WRITE://Write a word from a specific location in memory to the screen.
                        System.out.printf("WRITE from index %02d : ", operand);
                        System.out.printf("%d%n", array[operand]);
                        break;

                    // load/store operations
                    case LOAD: //Load a word from a specific location in memory into the accumulator.
                    accumulator = array[operand];
                    break;

                    case STORE: //Store a word from the accumulator into a specific location in memory.
                        array[operand] = accumulator;
                        break;

                    // arithmetic operations
                    case ADD: //Add a word from a specific location in memory to the word in the accumulator
                    accumulator += array[operand];
                    break;

                    case SUBTRACT: //Subtract a word from a specific location in memory from the word in the accumulator
                        accumulator -= array[operand];
                        break;

                    case DIVIDE: //Divide a word from a specific location in memory into the word in the accumulator
                        accumulator = array[operand] / accumulator;
                        break;

                    case MULTIPLY: //Multiply a word from a specific location in memory by the word in the accumulator
                        accumulator *= array[operand];
                        break;

                    // transfer of control operations
                    case BRANCH: //Branch to a specific location in memory
                        i = operand - 1; // subtracting 1 so that it cancels with the for loop increment
                        break;

                    case BRANCHNEG: //Branch to a specific location in memory if the accumulator is negative
                        if(accumulator < 0)
                            i = operand - 1;
                        break;

                    case BRANCHZERO: //Branch to a specific location in memory if the accumulator is zero
                        if(accumulator == 0)
                            i = operand - 1;
                        break;

                    //Ex 7.36 (a)
                    case SENTINEL_NEG:
                        int x = 0;
                        int sum = 0;
                        System.out.println("======== Exercise 7.36 (a) ========");System.out.
                        println();
                        for(int j = 0; j < 10; j++){
                            System.out.printf("READ at index %02d : ", operand + x);
                            array[operand + x] = input.nextInt();
                            if(array[operand + x] > 0){
                                sum += array[operand + x];
                                x++;
                            }
                            else{
                                sum += array[operand + x];
                                break;
                            }
                        }
                        array[operand + x + 1] = sum;
                        System.out.printf("SUM  at index %02d : %d", (operand + x + 1), sum);
                        break;

                    //Ex 7.36 (b)
                    case COUNTER_7:
                        int y = 0;
                        int sum2 = 0;
                        int average = 0;
                        System.out.println("======== Exercise 7.36 (b) ========");System.out.
                        println();
                        for(int j = 0; j < 7; j++){
                            System.out.printf("READ at index %02d : ", operand + y);
                            array[operand + y] = input.nextInt();
                            sum2 += array[operand + y];
                            y++;
                        }
                        average = sum2 / 7;
                        array[operand + y] = average;
                        System.out.printf("AVERAGE at index %02d : %d", (operand + y), average);
                        break;

                    case COMPARE:
                        int count;
                        int f = 2;
                        System.out.println("======== Exercise 7.36 (c) ========");System.out.
                        println();
                        System.out.printf("READ at index %02d : ", operand);
                        array[operand] = input.nextInt();
                        count = array[operand];
                        System.out.printf("READ at index %02d : ", operand + 1);
                        array[operand + 1] = input.nextInt();
                        int greatest = array[operand + 1];

                        for(int s = 0; s < count - 1; s++){
                            System.out.printf("READ at index %02d : ", operand + f);
                            array[operand + f] = input.nextInt();
                            if(array[operand + f] > greatest){
                                greatest = array[operand + f];
                            }
                            f++;
                        }
                        array[operand + f] = greatest;
                        System.out.printf("GREATEST at index %02d : %d", (operand + f), greatest
                        );
                        break;

                    case HALT: //Halt. The program has completed its task
                        found = true;
                        break;

                    default:
                        found = true;
                        System.out.printf("Operation code %d not found, %s%n", opcode, "the
                        program will end");
                }
                //if HALT or default is found, the program ends
                if(found == true)
                break;
            }

            //Output
            System.out.printf("%n%s%n%n", "============= REGISTERS =============");
            System.out.printf("Accumulator : %d%n", accumulator);
            System.out.printf("Instruction Register : %d%n", instructionReg);
            System.out.printf("Instruction Counter : %02d%n", i);
            System.out.printf("Operation Code : %d%n", opcode);
            System.out.printf("Operand Code : %02d%n", operand);

            //display the array (memory dump of Simpletron)
            System.out.printf("%n============ %s ============%n%n","MEMORY DUMP");
            System.out.print("MEMORY     0");
            for(int v = 1; v < 10; v++){
                System.out.printf("%10d", v);
            }
            System.out.println();

            for(int w = 0; w < array.length; w++){

                if(w % 10 == 0){
                    System.out.println();
                    System.out.printf("%d0", w / 10);
                    }
                System.out.printf("%10d", array[w]);
                }

            System.out.printf("%n%n%s%n", "=====================================");
            input.close();
        }

    }
}
```