**CSC 3101 - FALL 2016**

**Lab Assignment 7**

Please read the requirements carefully. Any missing point will result in a grade reduction. This assignment has to be submitted to your **blackboard in class** including the **Word file**. Hard copy is **not accepted**.

**Question:** **(100 pts.)**

Write a program in Easy68K to fulfill the following functions:

1. Read in two numbers from keyboard with instructions, one for **index**, one for the **size** of the array; (40 pts.)
2. Display the number in [**index % size**] inthe initialized array [2 3 8 1 6 5 4], starting from ‘0’. (60 pts.)

**Example:**

*Input the size of the array: 7*

*Input the index to display: 9*

(then 9%7=2, display the number in the array[2], which is 8)

*The value is: 8*

This should work for all cases in your program. Paste your code (not picture of code) and screenshots in this file.

\*-----------------------------------------------------------

\* Title : Lab 7

\* Written by : Caleb Latimer

\* Date : 10/26/2016

\* Description:

\*-----------------------------------------------------------

ORG $1000

START: ; first instruction of program

\*input\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*read a number through keyboard\*

MOVE.B #14,D0 ; task 13: display a string without CRLF

LEA M0,A1 ; assign address of MES to A1

TRAP #15

MOVE.B #4, D0 ; task 4: load number to D1.L

TRAP #15

MOVE.B D1,D2

CLR D1

MOVE.B #14,D0 ; task 13: display a string without CRLF

LEA M1,A1 ; assign address of MES to A1

TRAP #15

MOVE.B #4,D0 ; task 3: display signed number in D1.L

TRAP #15

MOVE.B #14,D0 ; newline

LEA NEWLINE,A1

TRAP #15

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*modulo\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MOVE.L D1,D3 ; store input number in D3.L

CLR.L D1 ; clear D1

DIVU D2,D3 ; compute modulo and remainder of D3/Y

; remainder is stored in higher bits of D3.L

; quotient is stored in lower bits of D3.L

SWAP D3 ; swap content of lower bits and higher bits

; now, modulo is stored in the lower bit of D3.L

MOVE.W D3,D1 ; move modulo to D1

MOVE.B #14,D0 ; display MES2

LEA M2,A1

TRAP #15

MOVE.B #3,D0 ; display modulo in D1.L

TRAP #15

MOVE.B #14,D0 ; newline

LEA NEWLINE,A1

TRAP #15

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*array\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*random access of an array\*

MOVE.B #14,D0 ; task 13: display a string without CRLF

LEA M3,A1 ; assign address of MES to A1

TRAP #15

LEA MYARRAY,A2 ; get address of SOMEARRAY

MULS #2,D1

ADD.W D1,A2 ; fourth element

; since the array is initialized with DC.W, each element takes 2 bytes

; so, the address of the 4th element= address of the 1st element + (2 bytes)\*(4-1)

MOVE.W (A2),D1 ; get SOMEARRAY[3]

MOVE.B #3,D0 ; display SOMEARRAY[3]

TRAP #15

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SIMHALT

\* Put variables and constants here

NEWLINE DC.B $D,$A,0

M0 DC.B 'Input the size of the array: ',0 ; Here, 0 indicates the end of a string, which is similar to '\0' in c-style strings

M1 DC.B 'Input the index to display: ',0

Y DC.W 10

M2 DC.B 'The modulo is: ',0

M3 DC.B 'The number is: ',0

MYARRAY DC.W 2,3,8,1,6,5,4 ; SOMEARRAY is similar to a pointer

; int SOMEARRAY[4]={3,4,2,8};

END START ; last line of source

