ASSIGNMENT

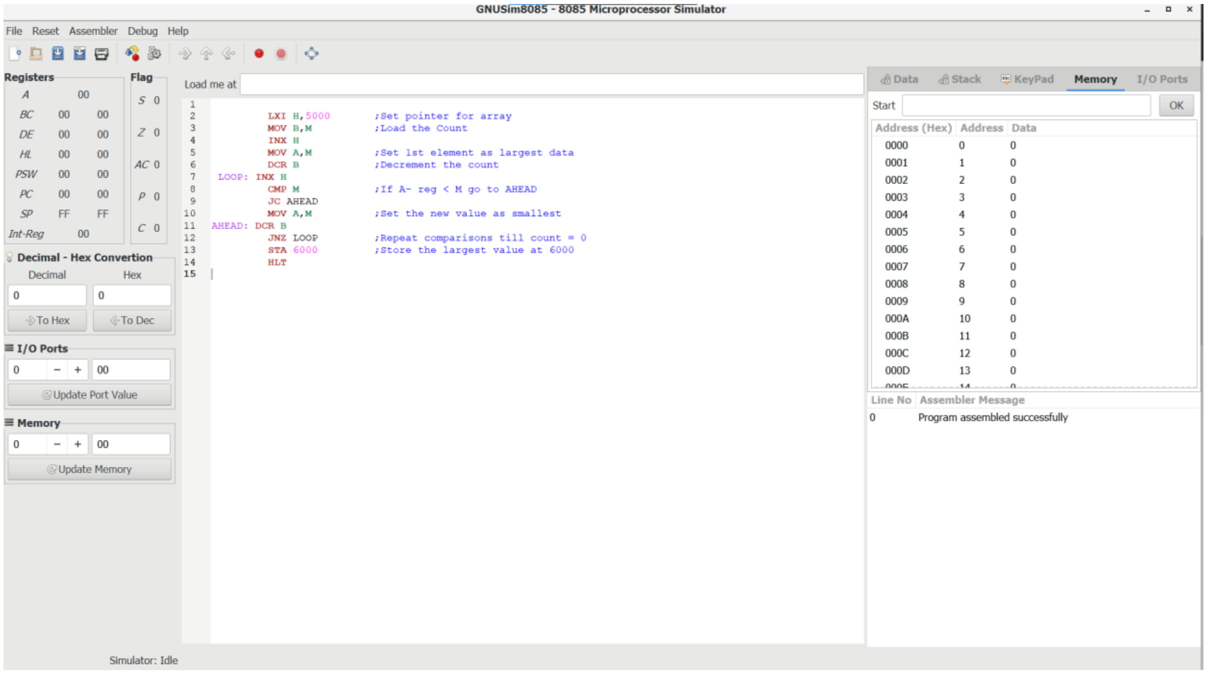
ADMISSION :146202-Mugambi Rintaugu and 138386 -FIRDAWS ALNUUR

Part One:

a.) Smallest number in an array of data

```
       LXI H,5000      ;Set pointer for array
        MOV B,M          ;Load the Count
        INX H
        MOV A,M          ;Set 1st element as largest data
        DCR B           ;Decrement the count
 LOOP: INX H
        CMP M           ;If A- reg < M go to AHEAD
        JC AHEAD
        MOV A,M          ;Set the new value as smallest
AHEAD: DCR B
        JNZ LOOP         ;Repeat comparisons till count = 0
        STA 6000        ;Store the largest value at 6000
        HLT
```

Code:



```
        LXI H,5000      ;Set pointer for array
        MOV B,M         ;Load the Count
        INX H
        MOV A,M         ;Set 1st element as largest data
        DCR B           ;Decrement the count
 LOOP:  INX H
        CMP M           ;If A- reg < M go to AHEAD
        JC AHEAD
        MOV A,M         ;Set the new value as smallest
AHEAD:  DCR B
        JNZ LOOP        ;Repeat comparisons till count = 0
        STA 6000        ;Store the largest value at 6000
        HLT
```

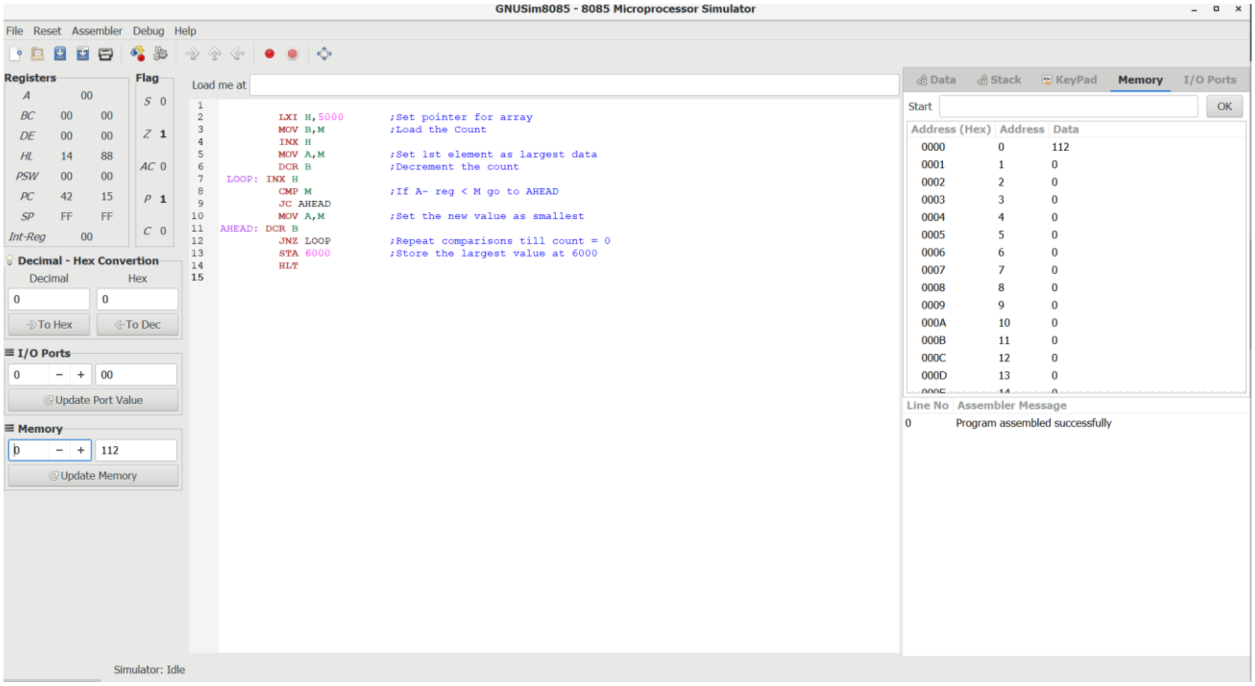
Registers		
A	00	
BC	00	00
DE	00	00
HL	00	00
PSW	00	00
PC	00	00
SP	FF	FF
Int-Reg	00	

Flag	
S	0
Z	0
AC	0
P	0
C	0

Address (Hex)	Address	Data
0000	0	0
0001	1	0
0002	2	0
0003	3	0
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0
000B	11	0
000C	12	0
000D	13	0

Line No	Assembler Message
0	Program assembled successfully

Results:



```
        LXI H,5000      ;Set pointer for array
        MOV B,M         ;Load the Count
        INX H
        MOV A,M         ;Set 1st element as largest data
        DCR B           ;Decrement the count
 LOOP:  INX H
        CMP M           ;If A- reg < M go to AHEAD
        JC AHEAD
        MOV A,M         ;Set the new value as smallest
AHEAD:  DCR B
        JNZ LOOP        ;Repeat comparisons till count = 0
        STA 6000        ;Store the largest value at 6000
        HLT
```

Registers		
A	00	
BC	00	00
DE	00	00
HL	14	88
PSW	00	00
PC	42	15
SP	FF	FF
Int-Reg	00	

Flag	
S	0
Z	1
AC	0
P	1
C	0

Address (Hex)	Address	Data
0000	0	112
0001	1	0
0002	2	0
0003	3	0
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0
000B	11	0
000C	12	0
000D	13	0

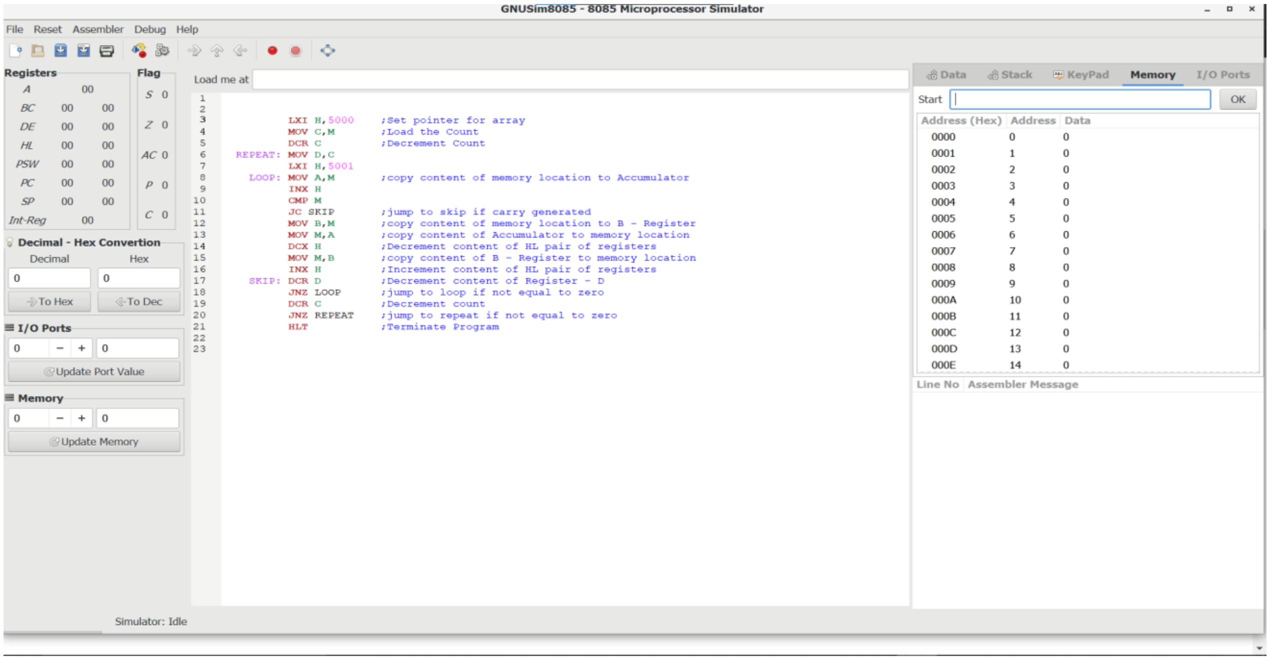
Line No	Assembler Message
0	Program assembled successfully

b.) Arranging in ascending order

LXI H,5000   ;Set pointer for array

```
        MOV C,M       ;Load the Count
        DCR C         ;Decrement Count
REPEAT: MOV D,C
        LXI H,5001
  LOOP: MOV A,M       ;copy content of memory location to Accumulator
        INX H
        CMP M
        JC SKIP       ;jump to skip if carry generated
        MOV B,M       ;copy content of memory location to B - Register
        MOV M,A       ;copy content of Accumulator to memory location
        DCX H         ;Decrement content of HL pair of registers
        MOV M,B       ;copy content of B - Register to memory location
        INX H         ;Increment content of HL pair of registers
  SKIP: DCR D         ;Decrement content of Register - D
        JNZ LOOP      ;jump to loop if not equal to zero
        DCR C         ;Decrement count
        JNZ REPEAT    ;jump to repeat if not equal to zero
        HLT           ;Terminate Program
```

CODE:



Registers		
A	00	
BC	00	00
DE	00	00
HL	00	00
PSW	00	00
PC	00	00
SP	00	00
Int-Reg	00	

Flag	
S	0
Z	0
AC	0
P	0
C	0

Decimal: 0 Hex: 0

I/O Ports: 0 − + 0

Memory: 0 − + 0

Load me at

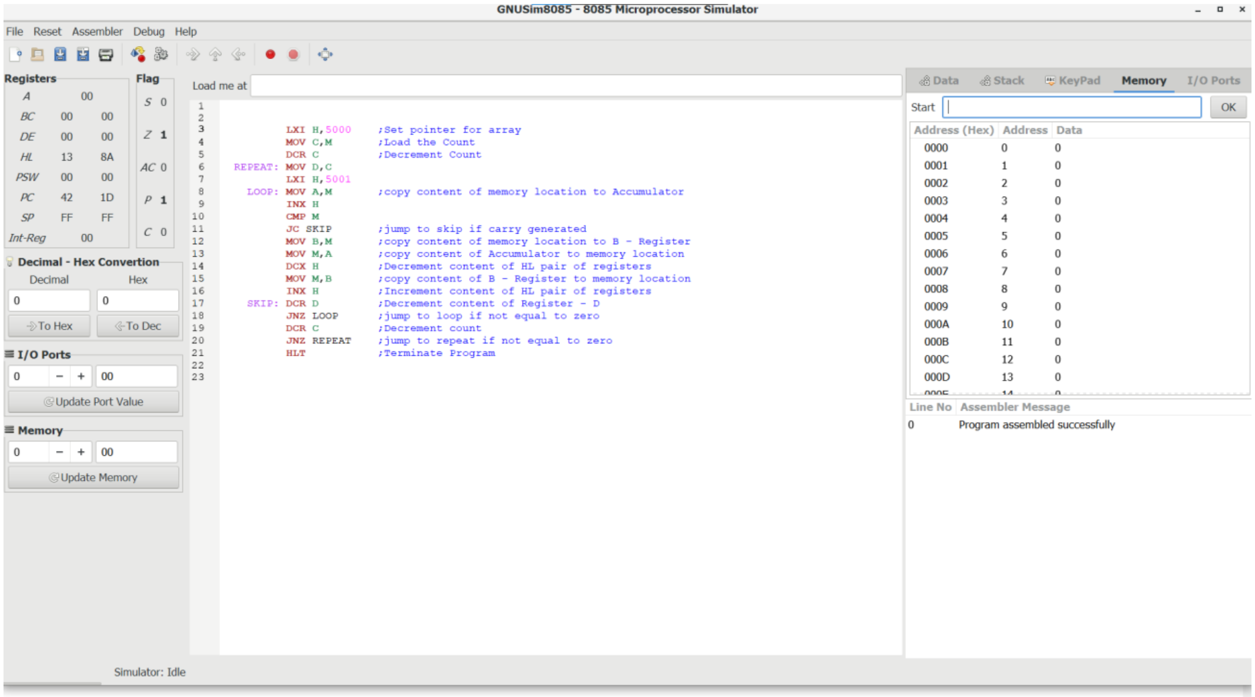
```
          LXI H,5000    ;Set pointer for array
          MOV C,M       ;Load the Count
          DCR C         ;Decrement Count
  REPEAT: MOV D,C
          LXI H,5001
    LOOP: MOV A,M       ;copy content of memory location to Accumulator
          INX H
          CMP M
          JC SKIP       ;jump to skip if carry generated
          MOV B,M       ;copy content of memory location to B - Register
          MOV M,A       ;copy content of Accumulator to memory location
          DCX H         ;Decrement content of HL pair of registers
          MOV M,B       ;copy content of B - Register to memory location
          INX H         ;Increment content of HL pair of registers
    SKIP: DCR D         ;Decrement content of Register - D
          JNZ LOOP      ;jump to loop if not equal to zero
          DCR C         ;Decrement count
          JNZ REPEAT    ;jump to repeat if not equal to zero
          HLT           ;Terminate Program
```

Address (Hex)	Address	Data
0000	0	0
0001	1	0
0002	2	0
0003	3	0
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0
000B	11	0
000C	12	0
000D	13	0
000E	14	0

Line No Assembler Message

Simulator: Idle

results:



Registers		
A	00	
BC	00	00
DE	00	00
HL	13	8A
PSW	00	00
PC	42	1D
SP	FF	FF
Int-Reg	00	

Flag	
S	0
Z	1
AC	0
P	1
C	0

Decimal: 0 Hex: 0

I/O Ports: 0 − + 00

Memory: 0 − + 00

Load me at

```
          LXI H,5000    ;Set pointer for array
          MOV C,M       ;Load the Count
          DCR C         ;Decrement Count
  REPEAT: MOV D,C
          LXI H,5001
    LOOP: MOV A,M       ;copy content of memory location to Accumulator
          INX H
          CMP M
          JC SKIP       ;jump to skip if carry generated
          MOV B,M       ;copy content of memory location to B - Register
          MOV M,A       ;copy content of Accumulator to memory location
          DCX H         ;Decrement content of HL pair of registers
          MOV M,B       ;copy content of B - Register to memory location
          INX H         ;Increment content of HL pair of registers
    SKIP: DCR D         ;Decrement content of Register - D
          JNZ LOOP      ;jump to loop if not equal to zero
          DCR C         ;Decrement count
          JNZ REPEAT    ;jump to repeat if not equal to zero
          HLT           ;Terminate Program
```

Address (Hex)	Address	Data
0000	0	0
0001	1	0
0002	2	0
0003	3	0
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0
000B	11	0
000C	12	0
000D	13	0
000E	14	0

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

Part Two: Measuring Performance

1. **Formula for CPU execution time for a program**

This is to say the metric of measurement for performance is the inverse of execution time.

$$Performance\ of\ A = \frac{1}{Execution\ time\ of\ A}$$

2. **CPU performance equation**

Performance equation analyzes execution time as a product of three factors; Instruction Count (IC), Clocks Per Instruction (CPI) and Clock Time (CT)

$$CPU\ Time = IC \times CPI \times\ CT$$

Instruction Count: the total number of instruction executions involved in a program

Clocks Per Instruction: the average number of clock cycles per instruction for a program or program fragment.

Clock Time: the period of the clock that synchronizes the circuits in a processor

Given that Clock Time is the inverse of Clock Frequency, we can rewrite the above formula as:

$$CPU\ Time = \frac{IC \times CPI}{Clock\ Rate}$$

3. **Formula for CPU clock cycles required for a program**

The time between two adjacent pulses of the oscillator that sets the tempo of the computer processor.

$$CPU\ Clock\ Cycles = Instructions\ for\ a\ Program\ \times Average\ Clock\ Cycles\ Per\ Instruction$$

4. Comparing two PC's performance

Performance of a computer refers to how well it performs a presented task. This is affected by the response time, through put and execution time of the system.

Execution time is inversely proportional to performance hence the expression:

$$Performance = (\frac{1}{Execution\ time})$$

Given the above equation we are able to derive an equation for comparing the performance of two computers

$$n = \left(Performance\ of\ \frac{A}{B}\right) = \left(Execution\ Time\ of\ \frac{B}{A}\right)$$

Given two processors A and B, A will be considered faster if it has lesser execution time in comparison to B and will be said to be *n* times faster than B

Example:

Execution time on machine A: Execution Time of A = 1 second

Execution time on machine B: Execution Time of B = 10 seconds

$$n = \left(Performance\ of\ \frac{A}{B}\right) = \left(Execution\ Time\ of\ \frac{B}{A}\right)$$

$$n = \left(Performance\ of\ \frac{A}{B}\right) = \left(Execution\ Time\ of\ \frac{40}{4}\right)$$

$$n = \left(Performance\ of\ \frac{A}{B}\right) = 10$$

The performance of machine A is 10 times the performance of machine B when running this program, or: Machine A is said to be 10 times faster than machine B when running this program.