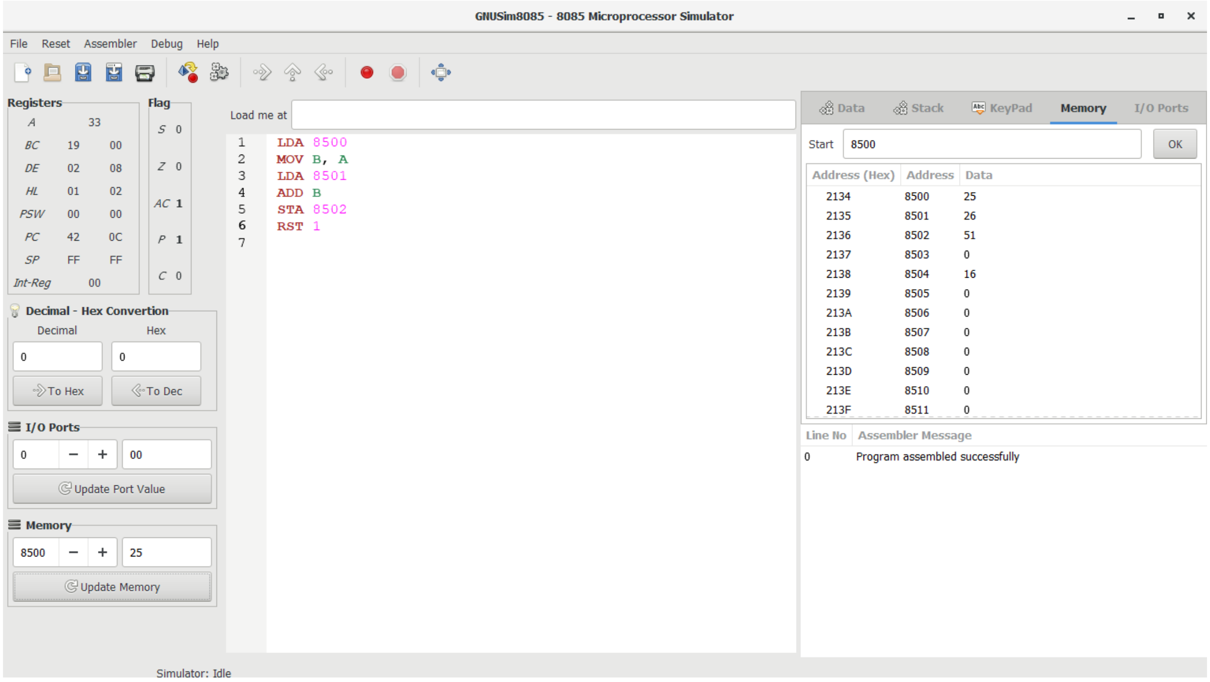
DAY 1: EXPERIMENTS -1 TO 11

1.8-BIT ADDITION



Registers		
A	33	
BC	19	00
DE	02	08
HL	01	02
PSW	00	00
PC	42	0C
SP	FF	FF
Int-Reg	00	

Flag	
S	0
Z	0
AC	1
P	1
C	0

Decimal: 0 Hex: 0

I/O Ports: 0 00

Memory: 8500 25

Load me at

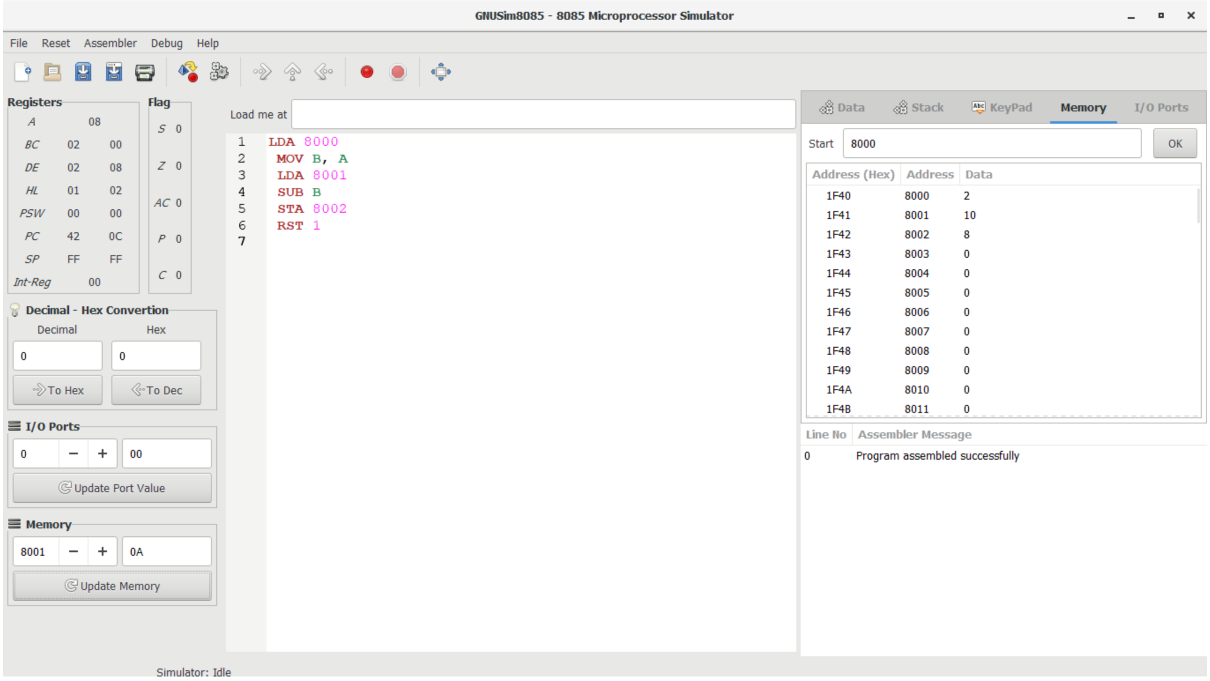
```
LDA 8500
MOV B, A
LDA 8501
ADD B
STA 8502
RST 1
```

Start 8500

Address (Hex)	Address	Data
2134	8500	25
2135	8501	26
2136	8502	51
2137	8503	0
2138	8504	16
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	0
213F	8511	0

Line No	Assembler Message
0	Program assembled successfully

2.8-BIT SUBTRACTION



Registers		
A	08	
BC	02	00
DE	02	08
HL	01	02
PSW	00	00
PC	42	0C
SP	FF	FF
Int-Reg	00	

Flag	
S	0
Z	0
AC	0
P	0
C	0

Decimal: 0 Hex: 0

I/O Ports: 0 00

Memory: 8001 0A

Load me at

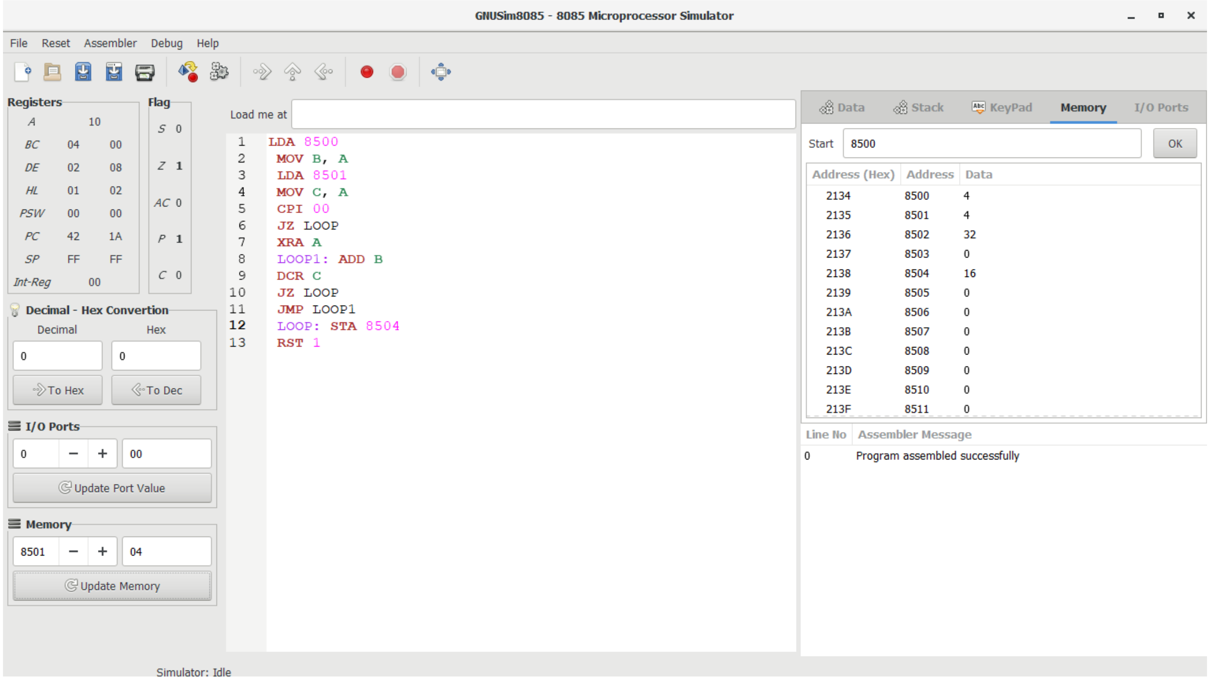
```
LDA 8000
MOV B, A
LDA 8001
SUB B
STA 8002
RST 1
```

Start 8000

Address (Hex)	Address	Data
1F40	8000	2
1F41	8001	10
1F42	8002	8
1F43	8003	0
1F44	8004	0
1F45	8005	0
1F46	8006	0
1F47	8007	0
1F48	8008	0
1F49	8009	0
1F4A	8010	0
1F4B	8011	0

Line No	Assembler Message
0	Program assembled successfully

3.8-BIT MULTIPLICATION


GNUSim8085 - 8085 Microprocessor Simulator

Registers		
A	10	
BC	04	00
DE	02	08
HL	01	02
PSW	00	00
PC	42	1A
SP	FF	FF
Int-Reg	00	

Flag	
S	0
Z	1
AC	0
P	1
C	0

Memory: 8501 — 04

```
LDA 8500
 MOV B, A
 LDA 8501
 MOV C, A
 CPI 00
 JZ LOOP
 XRA A
 LOOP1: ADD B
 DCR C
 JZ LOOP
 JMP LOOP1
 LOOP: STA 8504
 RST 1
```

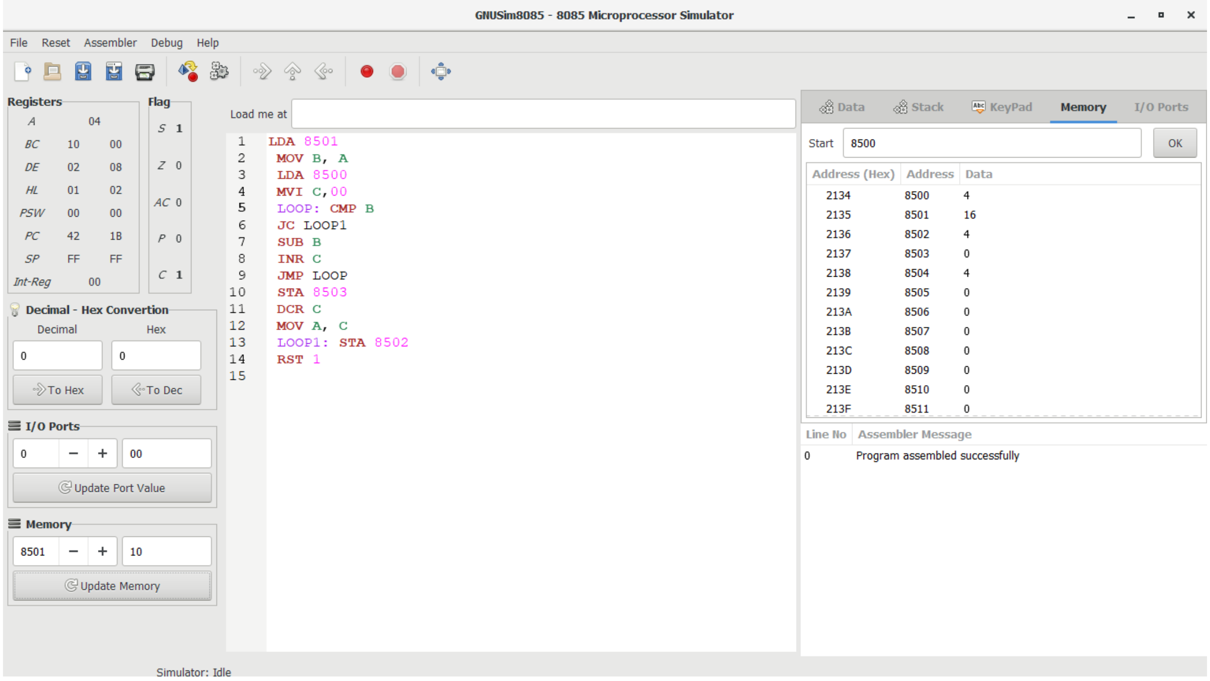
Start 8500

Address (Hex)	Address	Data
2134	8500	4
2135	8501	4
2136	8502	32
2137	8503	0
2138	8504	16
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	0
213F	8511	0

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

4.8-BIT DIVISION


GNUSim8085 - 8085 Microprocessor Simulator

Registers		
A	04	
BC	10	00
DE	02	08
HL	01	02
PSW	00	00
PC	42	1B
SP	FF	FF
Int-Reg	00	

Flag	
S	1
Z	0
AC	0
P	0
C	1

Memory: 8501 — 10

```
LDA 8501
 MOV B, A
 LDA 8500
 MVI C,00
 LOOP: CMP B
 JC LOOP1
 SUB B
 INR C
 JMP LOOP
 STA 8503
 DCR C
 MOV A, C
 LOOP1: STA 8502
 RST 1
```

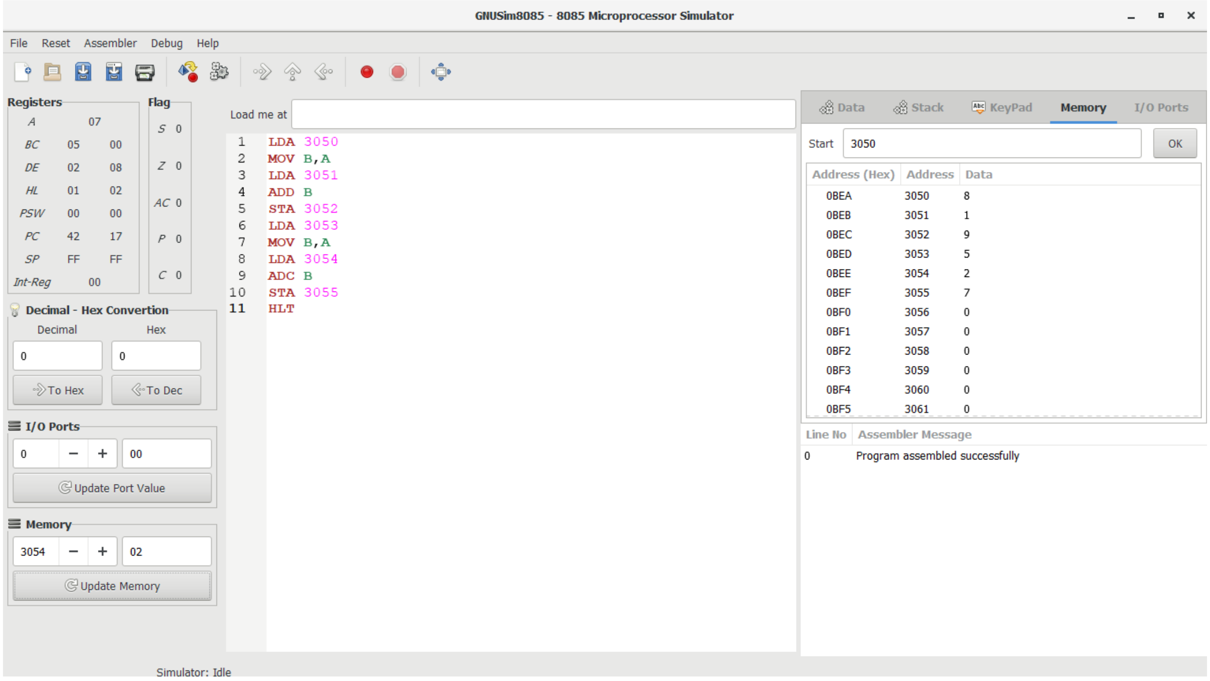
Start 8500

Address (Hex)	Address	Data
2134	8500	4
2135	8501	16
2136	8502	4
2137	8503	0
2138	8504	4
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	0
213F	8511	0

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

5.16-BIT ADDITION


GNUSim8085 - 8085 Microprocessor Simulator
File Reset Assembler Debug Help
Data Stack KeyPad Memory I/O Ports
To Hex
To Dec
Update Port Value
Update Memory
OK

Registers

A	07	
BC	05	00
DE	02	08
HL	01	02
PSW	00	00
PC	42	17
SP	FF	FF
Int-Reg	00	

Flag

S	0
Z	0
AC	0
P	0
C	0

Decimal - Hex Convertion

Decimal: 0 Hex: 0

I/O Ports: 0 00

Memory: 3054 02

Load me at

```
LDA 3050
MOV B,A
LDA 3051
ADD B
STA 3052
LDA 3053
MOV B,A
LDA 3054
ADC B
STA 3055
HLT
```

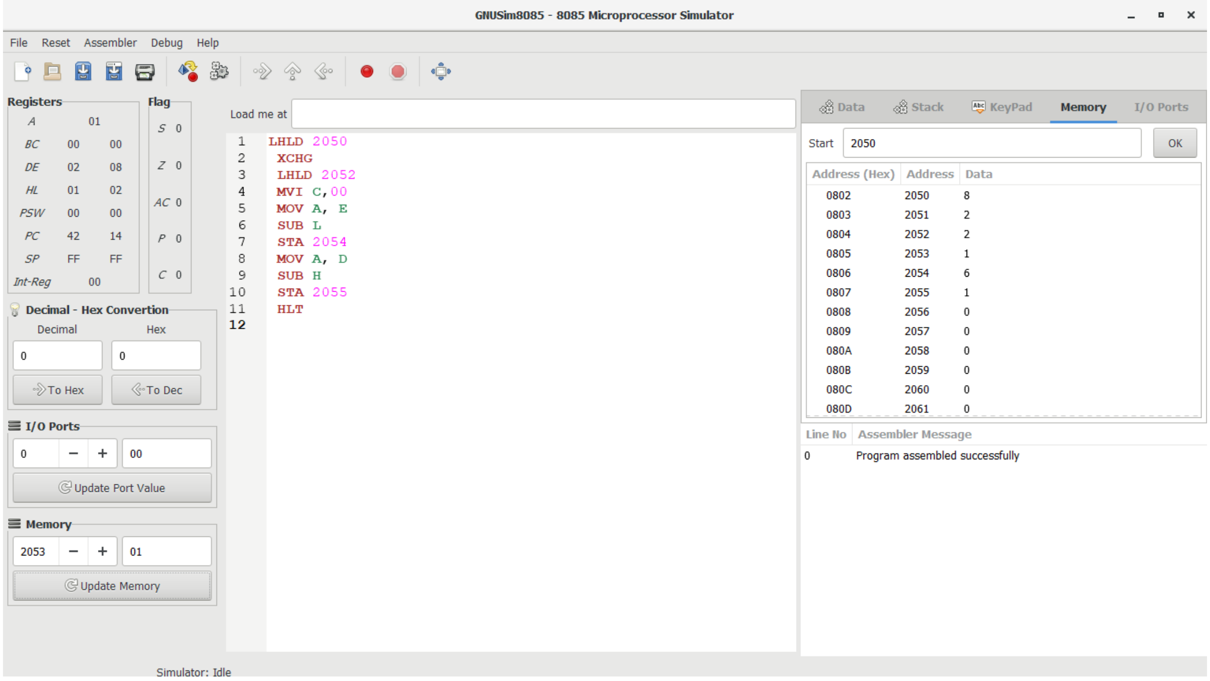
Start 3050

Address (Hex)	Address	Data
0BEA	3050	8
0BEB	3051	1
0BEC	3052	9
0BED	3053	5
0BEE	3054	2
0BEF	3055	7
0BF0	3056	0
0BF1	3057	0
0BF2	3058	0
0BF3	3059	0
0BF4	3060	0
0BF5	3061	0

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

6.16-BIT SUBTRACTION


GNUSim8085 - 8085 Microprocessor Simulator
File Reset Assembler Debug Help
Data Stack KeyPad Memory I/O Ports
To Hex
To Dec
Update Port Value
Update Memory
OK

Registers

A	01	
BC	00	00
DE	02	08
HL	01	02
PSW	00	00
PC	42	14
SP	FF	FF
Int-Reg	00	

Flag

S	0
Z	0
AC	0
P	0
C	0

Decimal - Hex Convertion

Decimal: 0 Hex: 0

I/O Ports: 0 00

Memory: 2053 01

Load me at

```
LHLD 2050
 XCHG
 LHLD 2052
 MVI C,00
 MOV A, E
 SUB L
 STA 2054
 MOV A, D
 SUB H
 STA 2055
 HLT
```

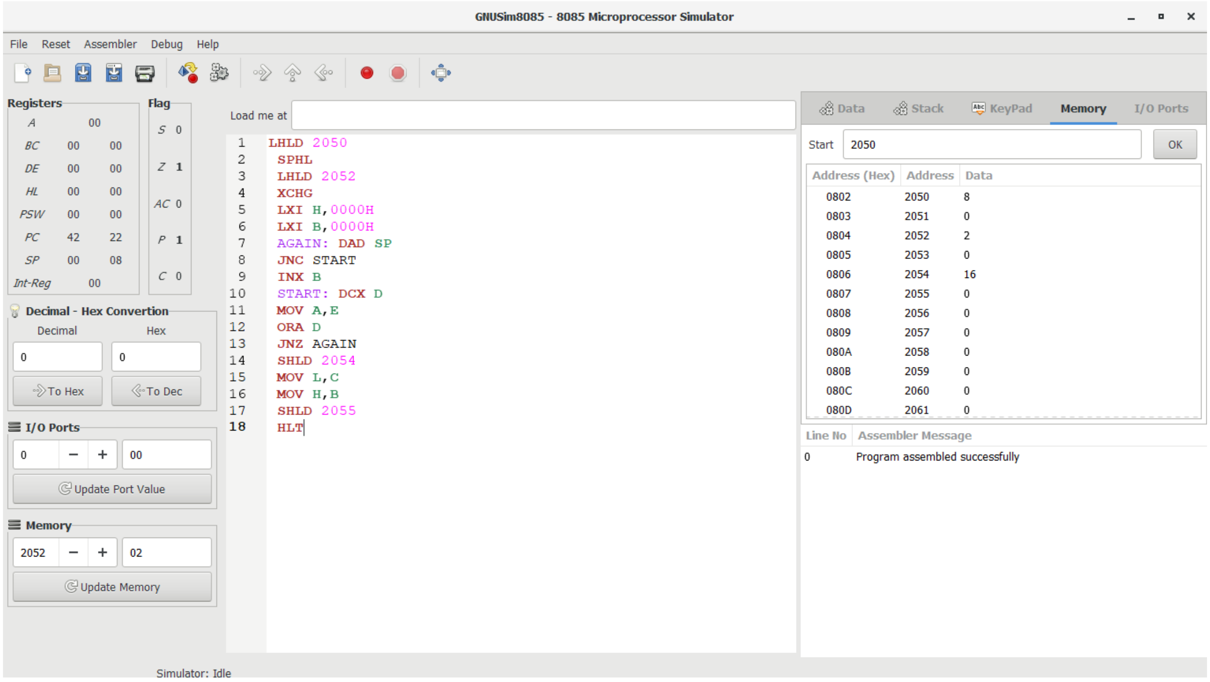
Start 2050

Address (Hex)	Address	Data
0802	2050	8
0803	2051	2
0804	2052	2
0805	2053	1
0806	2054	6
0807	2055	1
0808	2056	0
0809	2057	0
080A	2058	0
080B	2059	0
080C	2060	0
080D	2061	0

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

7.16-BIT MULTIPLICATION



Registers

A	00	
BC	00	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	22
SP	00	08
Int-Reg	00	

Flag

S	0
Z	1
AC	0
P	1
C	0

I/O Ports: 0 − + 00

Memory: 2052 − + 02

Decimal: 0 Hex: 0

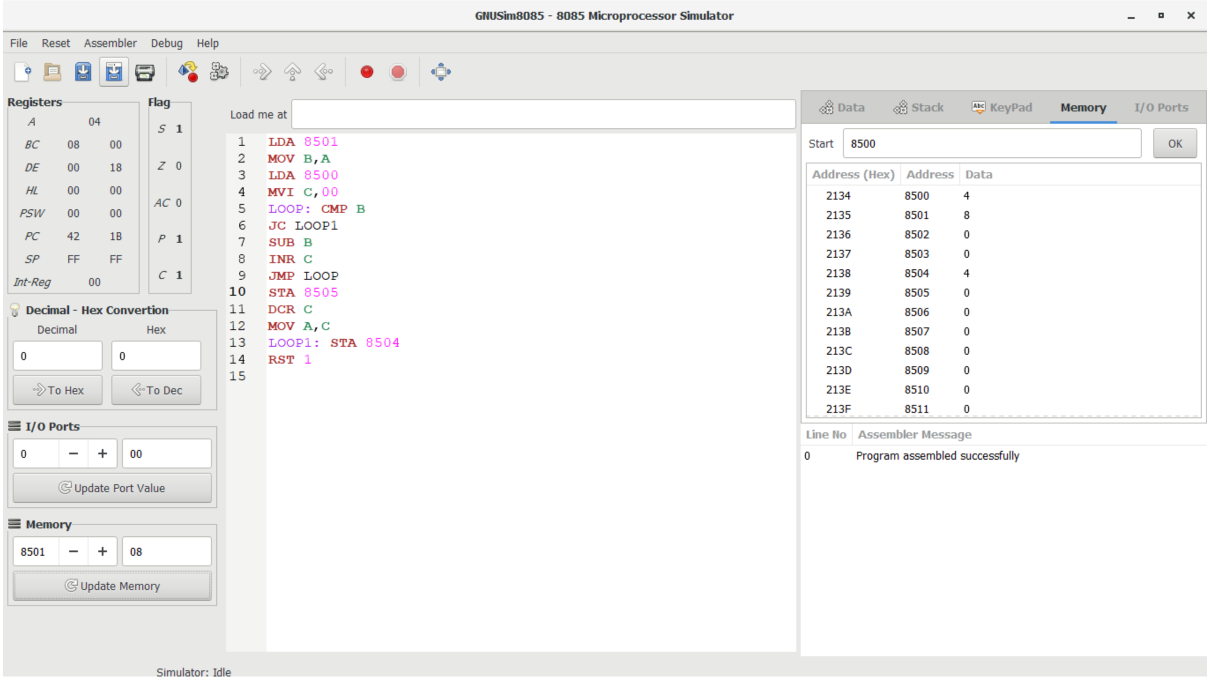
```
LHLD 2050
 SPHL
 LHLD 2052
 XCHG
 LXI H,0000H
 LXI B,0000H
 AGAIN: DAD SP
 JNC START
 INX B
 START: DCX D
 MOV A,E
 ORA D
 JNZ AGAIN
 SHLD 2054
 MOV L,C
 MOV H,B
 SHLD 2055
 HLT
```

Start 2050

Address (Hex)	Address	Data
0802	2050	8
0803	2051	0
0804	2052	2
0805	2053	0
0806	2054	16
0807	2055	0
0808	2056	0
0809	2057	0
080A	2058	0
080B	2059	0
080C	2060	0
080D	2061	0

Line No	Assembler Message
0	Program assembled successfully

8.16-BIT DIVISION



Registers

A	04	
BC	08	00
DE	00	18
HL	00	00
PSW	00	00
PC	42	1B
SP	FF	FF
Int-Reg	00	

Flag

S	1
Z	0
AC	0
P	1
C	1

I/O Ports: 0 − + 00

Memory: 8501 − + 08

Decimal: 0 Hex: 0

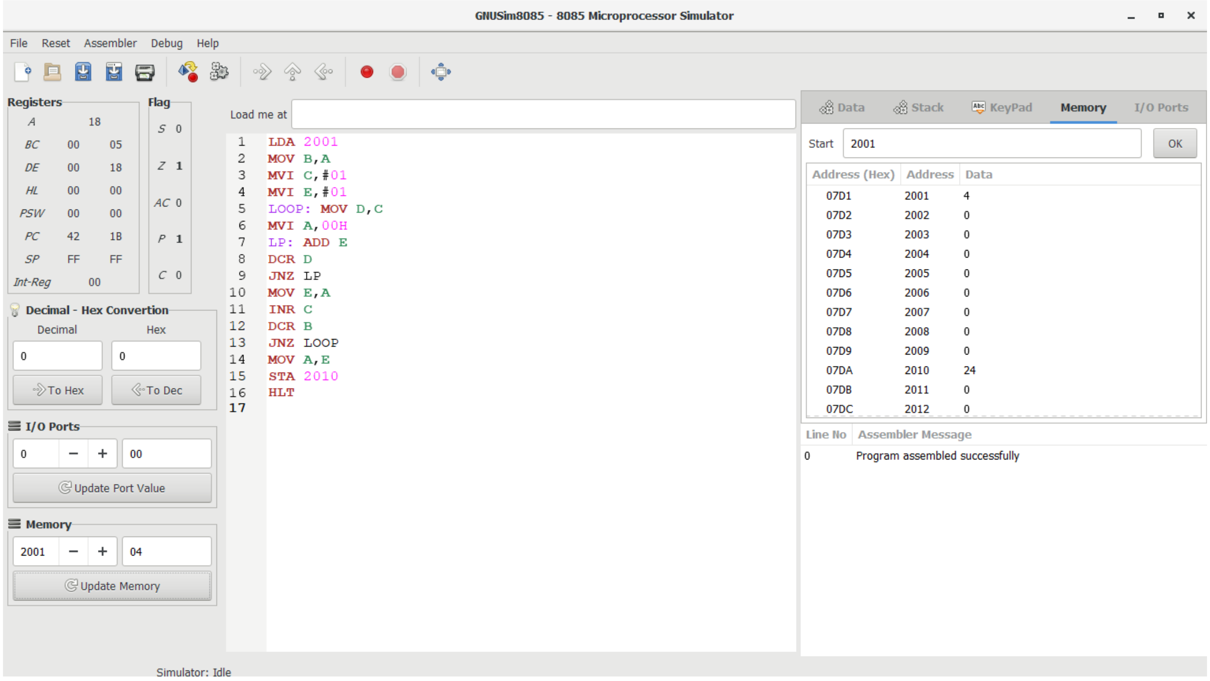
```
LDA 8501
MOV B,A
LDA 8500
MVI C,00
LOOP: CMP B
JC LOOP1
SUB B
INR C
JMP LOOP
STA 8505
DCR C
MOV A,C
LOOP1: STA 8504
RST 1
```

Start 8500

Address (Hex)	Address	Data
2134	8500	4
2135	8501	8
2136	8502	0
2137	8503	0
2138	8504	4
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	0
213F	8511	0

Line No	Assembler Message
0	Program assembled successfully

9 . FACTORIAL IN 8085


GNUSim8085 - 8085 Microprocessor Simulator
File Reset Assembler Debug Help
Data Stack KeyPad Memory I/O Ports
To Hex To Dec
Update Port Value
Update Memory
OK

Registers		
A	18	
BC	00	05
DE	00	18
HL	00	00
PSW	00	00
PC	42	1B
SP	FF	FF
Int-Reg	00	

Flag	
S	0
Z	1
AC	0
P	1
C	0

Decimal - Hex Convertion: Decimal 0, Hex 0

I/O Ports: 0, 00

Memory: 2001, 04

Load me at

```
LDA 2001
MOV B,A
MVI C,#01
MVI E,#01
LOOP: MOV D,C
MVI A,00H
LP: ADD E
DCR D
JNZ LP
MOV E,A
INR C
DCR B
JNZ LOOP
MOV A,E
STA 2010
HLT
```

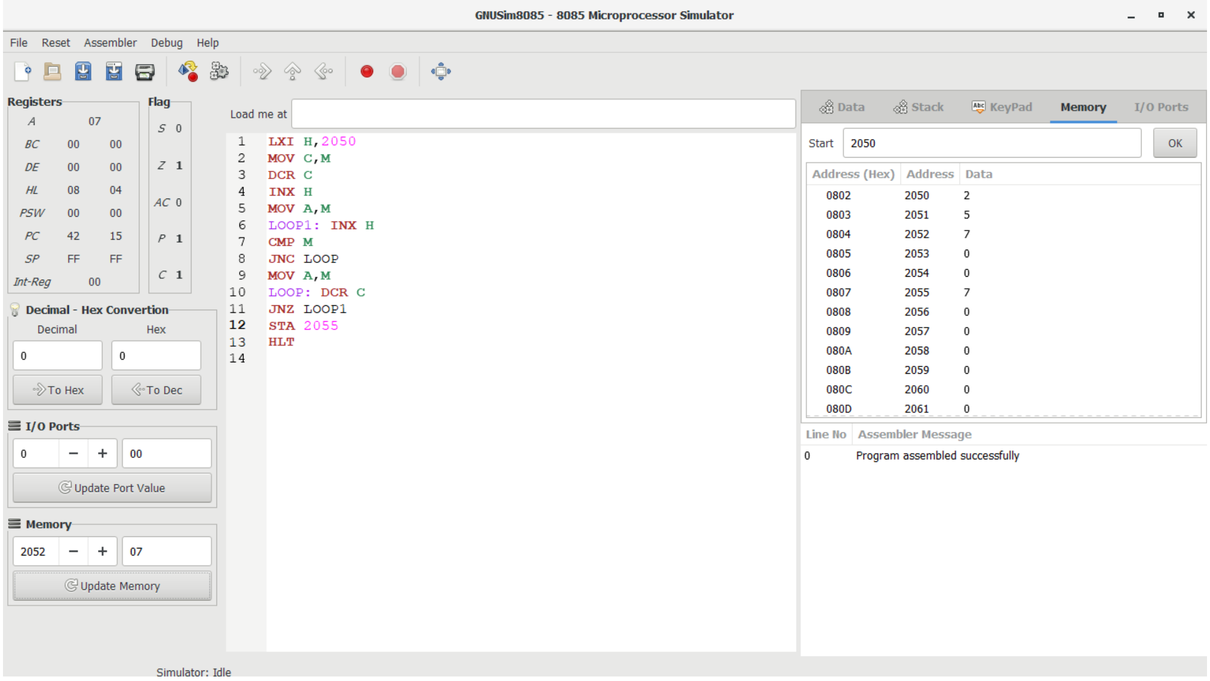
Start 2001

Address (Hex)	Address	Data
07D1	2001	4
07D2	2002	0
07D3	2003	0
07D4	2004	0
07D5	2005	0
07D6	2006	0
07D7	2007	0
07D8	2008	0
07D9	2009	0
07DA	2010	24
07DB	2011	0
07DC	2012	0

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

10.LARGEST NUMBER IN AN ARRAY


GNUSim8085 - 8085 Microprocessor Simulator
File Reset Assembler Debug Help
Data Stack KeyPad Memory I/O Ports
To Hex To Dec
Update Port Value
Update Memory
OK

Registers		
A	07	
BC	00	00
DE	00	00
HL	08	04
PSW	00	00
PC	42	15
SP	FF	FF
Int-Reg	00	

Flag	
S	0
Z	1
AC	0
P	1
C	1

Decimal - Hex Convertion: Decimal 0, Hex 0

I/O Ports: 0, 00

Memory: 2052, 07

Load me at

```
LXI H,2050
MOV C,M
DCR C
INX H
MOV A,M
LOOP1: INX H
CMP M
JNC LOOP
MOV A,M
LOOP: DCR C
JNZ LOOP1
STA 2055
HLT
```

Start 2050

Address (Hex)	Address	Data
0802	2050	2
0803	2051	5
0804	2052	7
0805	2053	0
0806	2054	0
0807	2055	7
0808	2056	0
0809	2057	0
080A	2058	0
080B	2059	0
080C	2060	0
080D	2061	0

Line No	Assembler Message
0	Program assembled successfully

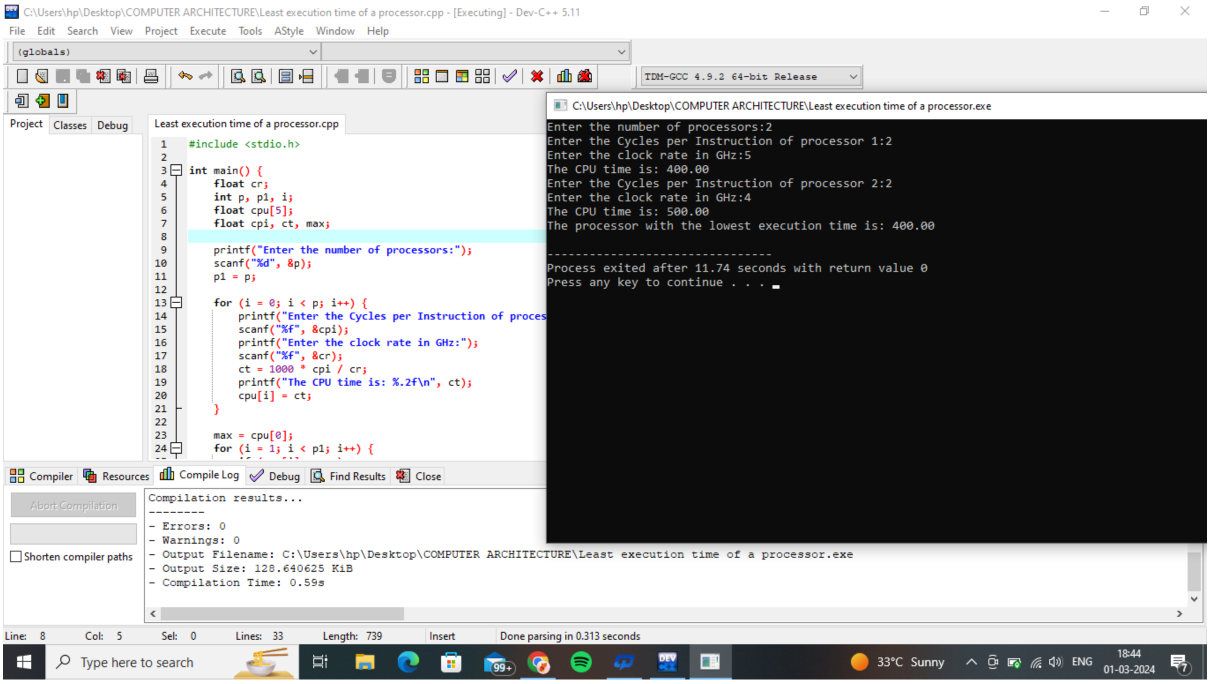
Simulator: Idle

11.LEAST EXCECUTION TIME

```
#include <stdio.h>


int main() {
   float cr;
```

```
int p, p1, i;
float cpu[5];
float cpi, ct, max;

printf("Enter the number of processors:");
scanf("%d", &p);
p1 = p;

for (i = 0; i < p; i++) {
    printf("Enter the Cycles per Instruction of processor %d:", i+1);
    scanf("%f", &cpi);
    printf("Enter the clock rate in GHz:");
    scanf("%f", &cr);
    ct = 1000 * cpi / cr;
    printf("The CPU time is: %.2f\n", ct);
    cpu[i] = ct;
}

max = cpu[0];
for (i = 1; i < p1; i++) {
    if (cpu[i] < max)
        max = cpu[i];
}
```

```
    printf("The processor with the lowest execution time is: %.2f\n ", max);


    return 0;
}
```



C:\Users\hp\Desktop\COMPUTER ARCHITECTURE\Least execution time of a processor.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
TDM-GCC 4.9.2 64-bit Release
Project Classes Debug
Least execution time of a processor.cpp
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
□ Shorten compiler paths
Type here to search
33°C Sunny
ENG
18:44
01-03-2024

```
#include <stdio.h>

int main() {
    float cr;
    int p, p1, i;
    float cpu[5];
    float cpi, ct, max;

    printf("Enter the number of processors:");
    scanf("%d", &p);
    p1 = p;

    for (i = 0; i < p; i++) {
        printf("Enter the Cycles per Instruction of proces
        scanf("%f", &cpi);
        printf("Enter the clock rate in GHz:");
        scanf("%f", &cr);
        ct = 1000 * cpi / cr;
        printf("The CPU time is: %.2f\n", ct);
        cpu[i] = ct;
    }

    max = cpu[0];
    for (i = 1; i < p1; i++) {
```

C:\Users\hp\Desktop\COMPUTER ARCHITECTURE\Least execution time of a processor.exe

```
Enter the number of processors:2
Enter the Cycles per Instruction of processor 1:2
Enter the clock rate in GHz:5
The CPU time is: 400.00
Enter the Cycles per Instruction of processor 2:2
Enter the clock rate in GHz:4
The CPU time is: 500.00
The processor with the lowest execution time is: 400.00

--------------------------------
Process exited after 11.74 seconds with return value 0
Press any key to continue . . .
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hp\Desktop\COMPUTER ARCHITECTURE\Least execution time of a processor.exe
- Output Size: 128.640625 KiB
- Compilation Time: 0.59s
```

Line: 8 Col: 5 Sel: 0 Lines: 33 Length: 739 Insert Done parsing in 0.313 seconds