# M.I.T. LAB Assignment – 02

# U19CS012

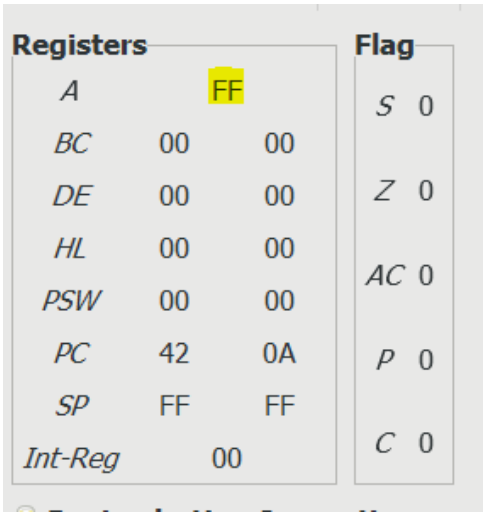
(1) Write a program for one's complement of 8-bit number.

Notepad Code:

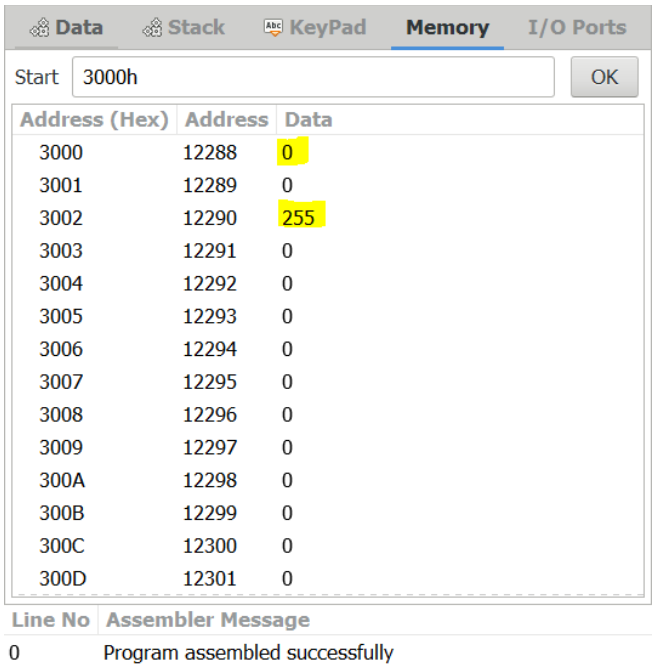
```
; Q-(1) Write a program for one's complement of 8-bit number

MVI A,00
; Load the Data at Location 3000 to Accumulator
LDA 3000h                    ; [A] <- [3000]
; CMA -> Complement the Contents of Accumulator
CMA                          ; [A] <- Complement([A])
; Store the Result at Location 3002
STA 3002h                    ; [3002] <- [A]

hlt
```

Registers and Memory:

Eg: (1) Let A = 00H = $(0000\ 0000)_2$ = 0 => A' = FFH = $(1111\ 1111)_2$ = 255



| Registers | | | Flag | |
|---|---|---|---|---|
| A | FF | | S | 0 |
| BC | 00 | 00 | Z | 0 |
| DE | 00 | 00 | AC | 0 |
| HL | 00 | 00 | P | 0 |
| PSW | 00 | 00 | C | 0 |
| PC | 42 | 0A | | |
| SP | FF | FF | | |
| Int-Reg | 00 | | | |



Data | Stack | KeyPad | Memory | I/O Ports

Start 3000h OK

| Address (Hex) | Address | Data |
|---|---|---|
| 3000 | 12288 | 0 |
| 3001 | 12289 | 0 |
| 3002 | 12290 | 255 |
| 3003 | 12291 | 0 |
| 3004 | 12292 | 0 |
| 3005 | 12293 | 0 |
| 3006 | 12294 | 0 |
| 3007 | 12295 | 0 |
| 3008 | 12296 | 0 |
| 3009 | 12297 | 0 |
| 300A | 12298 | 0 |
| 300B | 12299 | 0 |
| 300C | 12300 | 0 |
| 300D | 12301 | 0 |

| Line No | Assembler Message |
|---|---|
| 0 | Program assembled successfully |

Eg: (2) Let A = 96H = $(1001\ 0110)_2$ = 150 => A' = 69H = $(0110\ 1001)_2$ = 105

**Registers**

| | | |
|---|---|---|
| A | 69 | |
| BC | 00 | 00 |
| DE | 00 | 00 |
| HL | 00 | 00 |
| PSW | 00 | 00 |
| PC | 42 | 0A |
| SP | FF | FF |
| Int-Reg | 00 | |

Data | Stack | KeyPad | **Memory** | I/O Ports

Start 3000h OK

| Address (Hex) | Address | Data |
|---|---|---|
| 3000 | 12288 | 150 |
| 3001 | 12289 | 0 |
| 3002 | 12290 | 105 |
| 3003 | 12291 | 0 |
| 3004 | 12292 | 0 |
| 3005 | 12293 | 0 |
| 3006 | 12294 | 0 |
| 3007 | 12295 | 0 |
| 3008 | 12296 | 0 |
| 3009 | 12297 | 0 |
| 300A | 12298 | 0 |
| 300B | 12299 | 0 |
| 300C | 12300 | 0 |
| 300D | 12301 | 0 |

| Line No | Assembler Message |
|---|---|
| 0 | Program assembled successfully |

(2) Write a program for two's complement of 8-bit number.

<u>Notepad Code:</u>

```
; Q-(2) Write a program for two's complement of 8-bit number.

MVI A,00
; Load the Data at Location 3000 to Accumulator
LDA 3000h                    ; [A] <- [3000]
; CMA -> Complement the Contents of Accumulator
CMA                          ; [A] <- Complement([A])
; Add 01 to One's Complement to Obtain Two's Complement
ADI 01                       ;  [A] <- [A] + 01
; Store the Result at Location 3004
STA 3004h                    ; [3004] <- Two's_Complement([A])

hlt
```

## Registers and Memory:

Eg: (1) Let A = 01H = $(0000\ 0001)_2$ = 1 => A' + 1 = FFH = $(1111\ 1111)_2$ = 255

| Registers | | | Flag | |
|---|---|---|---|---|
| A | FF | | S | 1 |
| BC | 00 | 00 | Z | 0 |
| DE | 00 | 00 | AC | 0 |
| HL | 00 | 00 | P | 1 |
| PSW | 00 | 00 | C | 0 |
| PC | 42 | 0C | | |
| SP | FF | FF | | |
| Int-Reg | 00 | | | |

Data | Stack | KeyPad | Memory | I/O Ports

Start 3000h OK

| Address (Hex) | Address | Data |
|---|---|---|
| 3000 | 12288 | 1 |
| 3001 | 12289 | 0 |
| 3002 | 12290 | 0 |
| 3003 | 12291 | 0 |
| 3004 | 12292 | 255 |
| 3005 | 12293 | 0 |
| 3006 | 12294 | 0 |
| 3007 | 12295 | 0 |
| 3008 | 12296 | 0 |
| 3009 | 12297 | 0 |
| 300A | 12298 | 0 |
| 300B | 12299 | 0 |
| 300C | 12300 | 0 |
| 300D | 12301 | 0 |

| Line No | Assembler Message |
|---|---|
| 0 | Program assembled successfully |

Eg: (2) Let A = 96H = $(1001\ 0110)_2$ = 150 => A' + 1 = 6AH = $(0110\ 1010)_2$ = 106

| Registers | | | Flag | |
|---|---|---|---|---|
| A | 6A | | S | 0 |
| BC | 00 | 00 | Z | 0 |
| DE | 00 | 00 | AC | 0 |
| HL | 00 | 00 | P | 1 |
| PSW | 00 | 00 | C | 0 |
| PC | 42 | 0C | | |
| SP | FF | FF | | |
| Int-Reg | 00 | | | |

Data | Stack | KeyPad | Memory | I/O Ports

Start 3000h OK

| Address (Hex) | Address | Data |
|---|---|---|
| 3000 | 12288 | 150 |
| 3001 | 12289 | 0 |
| 3002 | 12290 | 0 |
| 3003 | 12291 | 0 |
| 3004 | 12292 | 106 |
| 3005 | 12293 | 0 |
| 3006 | 12294 | 0 |
| 3007 | 12295 | 0 |
| 3008 | 12296 | 0 |
| 3009 | 12297 | 0 |
| 300A | 12298 | 0 |
| 300B | 12299 | 0 |
| 300C | 12300 | 0 |
| 300D | 12301 | 0 |

| Line No | Assembler Message |
|---|---|
| 0 | Program assembled successfully |

(3) Write an assembly language program that AND, OR and XOR together the contents of register B, C and E and place the result into memory location 3000H, 3001H and 3002H.

Notepad Code:

```
; Q-(3)
; 3000H <- (B&C&E), 3001H <- (B|C|E) and 3002H <- (B^C^E)
MVI A,00
; Intialize the Register B with 8-bit [Number 1] at Location 3004h
LDA 3004h
MOV B, A
; Intialize the Register C with 8-bit [Number 2] at Location 3006h
LDA 3006h
MOV C, A
; Intialize the Register E with 8-bit [Number 3] at Location 3008h
LDA 3008h
MOV E, A
; AND Operation
MOV A, B
ANA C
ANA E
STA 3000h
; OR Operation
MOV A, B
ORA C
ORA E
STA 3001h
; XOR Operation
MOV A, B
XRA C
XRA E
STA 3002h
hlt
```

## Registers and Memory:

Eg: (1) Let

B = 96H = $(1001\ 0110)_2$ = 150

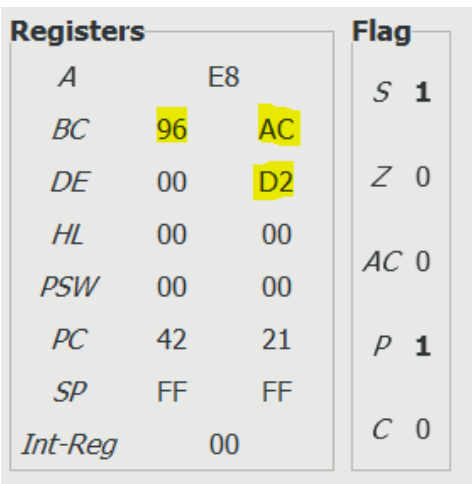
C = ACH = $(1010\ 1100)_2$ = 172

E = D2H = $(1101\ 0010)_2$ = 210

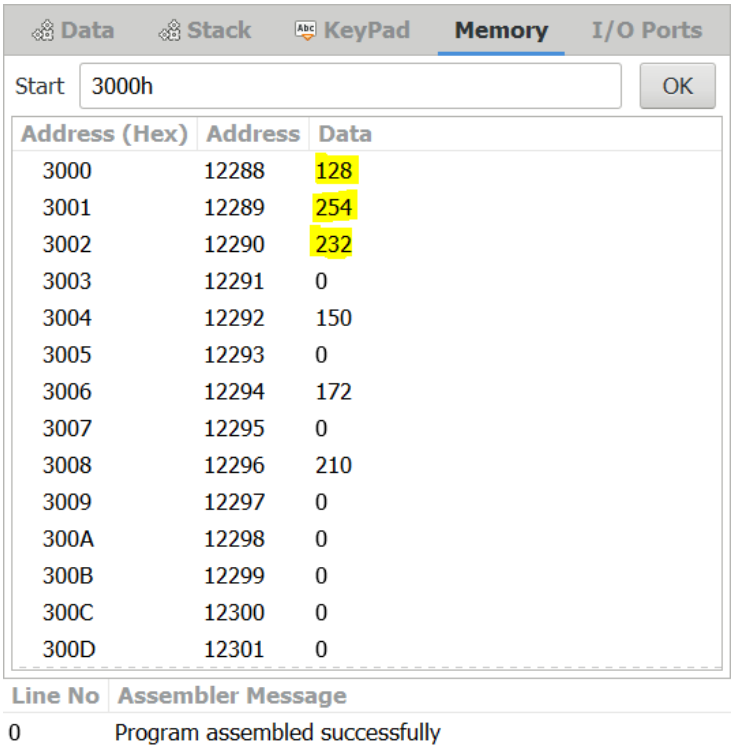
AND = (80)H = $(1000\ 0000)_2$ = 128

OR = (FE)H = $(1111\ 1110)_2$ = 254

XOR = (E8)H = $(1110\ 1000)_2$ = 232



| Registers | | | Flag | |
|---|---|---|---|---|
| A | E8 | | S | 1 |
| BC | 96 | AC | Z | 0 |
| DE | 00 | D2 | AC | 0 |
| HL | 00 | 00 | P | 1 |
| PSW | 00 | 00 | C | 0 |
| PC | 42 | 21 | | |
| SP | FF | FF | | |
| Int-Reg | 00 | | | |



Data | Stack | KeyPad | Memory | I/O Ports

Start 3000h OK

| Address (Hex) | Address | Data |
|---|---|---|
| 3000 | 12288 | 128 |
| 3001 | 12289 | 254 |
| 3002 | 12290 | 232 |
| 3003 | 12291 | 0 |
| 3004 | 12292 | 150 |
| 3005 | 12293 | 0 |
| 3006 | 12294 | 172 |
| 3007 | 12295 | 0 |
| 3008 | 12296 | 210 |
| 3009 | 12297 | 0 |
| 300A | 12298 | 0 |
| 300B | 12299 | 0 |
| 300C | 12300 | 0 |
| 300D | 12301 | 0 |

| Line No | Assembler Message |
|---|---|
| 0 | Program assembled successfully |

(4) Write a program to shift 8-bit no by three bits left. Assume data is in register C.

Notepad Code:

```
; Q-(4) Write a program to shift 8-bit no by three bits left. Assume data
; is in register C

MVI A,00
; Intialise the Register C with 8-Bit Number at Location 3000h
LDA 3000h
MOV C,A
; Rotate Accumulator Left [1st Time] [Accumulator has C's Data]
RAL
; Rotate Accumulator Left [2st Time]
RAL
; Rotate Accumulator Left [3st Time]
RAL
;Store the Results at Location 3002h
STA 3002h
hlt
```

<u>Registers and Memory:</u>

Eg: (1) Let C = 05H = $(0000\ 0101)_2$ = 5

Shift Left 1st Time = $(0000\ 1010)_2$ = 10

Shift Left 2nd Time = $(0001\ 0100)_2$ = 20

Shift Left 3rd Time = $(0010\ 1000)_2$= 40 = 28H

**Registers**

| | | |
|---|---|---|
| *A* | 28 | |
| *BC* | 00 | 05 |
| *DE* | 00 | 00 |
| *HL* | 00 | 00 |
| *PSW* | 00 | 00 |
| *PC* | 42 | 0D |
| *SP* | FF | FF |
| *Int-Reg* | 00 | |

Data | Stack | KeyPad | **Memory** | I/O Ports

Start 3000h OK

| Address (Hex) | Address | Data |
|---|---|---|
| 3000 | 12288 | 5 |
| 3001 | 12289 | 0 |
| 3002 | 12290 | 40 |
| 3003 | 12291 | 0 |
| 3004 | 12292 | 0 |
| 3005 | 12293 | 0 |
| 3006 | 12294 | 0 |
| 3007 | 12295 | 0 |
| 3008 | 12296 | 0 |
| 3009 | 12297 | 0 |
| 300A | 12298 | 0 |
| 300B | 12299 | 0 |
| 300C | 12300 | 0 |
| 300D | 12301 | 0 |

| Line No | Assembler Message |
|---|---|
| 0 | Program assembled successfully |

(5) Write a program to shift 8-bit data four bits right. Assume data is present in register C.

Notepad Code:

```
; Q-(5) Write a program to shift 8-bit no by three bits right. Assume data
; is in register C

MVI A,00
; Intialise the Register C with 8-Bit Number at Location 3000h
LDA 3000h
MOV C,A
; Rotate Accumulator Right [1st Time] [Accumulator has C's Data]
RAR
; Rotate Accumulator Right [2st Time]
RAR
; Rotate Accumulator Right [3st Time]
RAR
;Store the Results at Location 3002h
STA 3002h
hlt
```

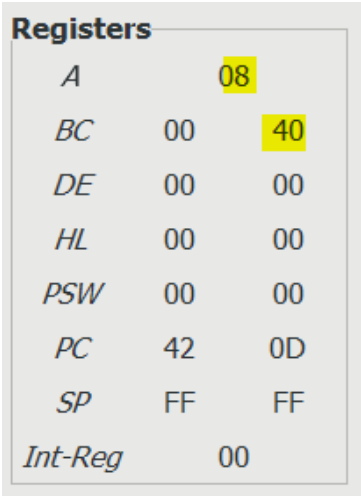
Registers and Memory:

Eg: (1) Let B = 05H = $(0100\ 0000)_2$ = 64 = 40H

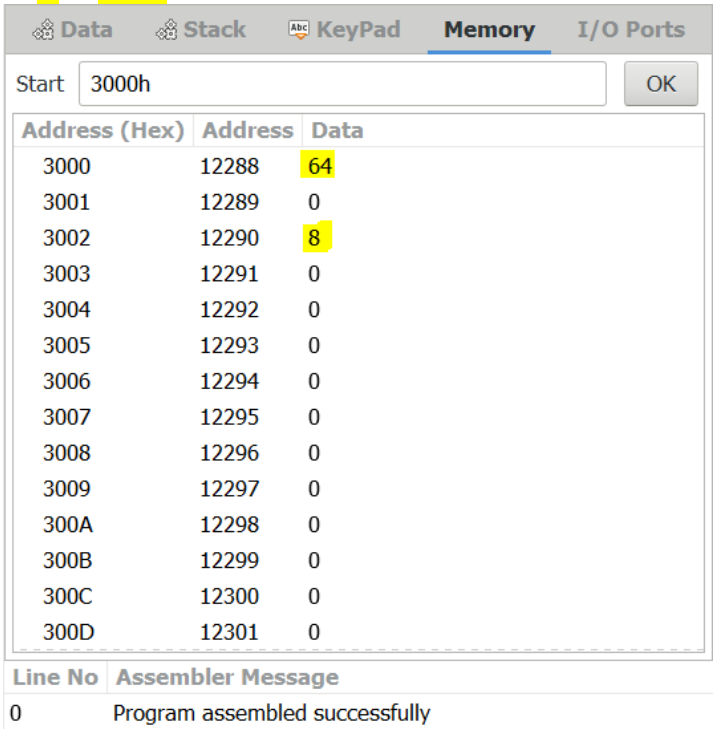
Shift Right 1st Time = $(0010\ 0000)_2$ = 32

Shift Right 2nd Time = $(0001\ 0000)_2$ = 16

Shift Right 3rd Time = $(0000\ 1000)_2$= 8 = 08H



| Registers | | |
|---|---|---|
| A | 08 | |
| BC | 00 | 40 |
| DE | 00 | 00 |
| HL | 00 | 00 |
| PSW | 00 | 00 |
| PC | 42 | 0D |
| SP | FF | FF |
| Int-Reg | 00 | |



Data | Stack | KeyPad | Memory | I/O Ports

Start 3000h OK

| Address (Hex) | Address | Data |
|---|---|---|
| 3000 | 12288 | 64 |
| 3001 | 12289 | 0 |
| 3002 | 12290 | 8 |
| 3003 | 12291 | 0 |
| 3004 | 12292 | 0 |
| 3005 | 12293 | 0 |
| 3006 | 12294 | 0 |
| 3007 | 12295 | 0 |
| 3008 | 12296 | 0 |
| 3009 | 12297 | 0 |
| 300A | 12298 | 0 |
| 300B | 12299 | 0 |
| 300C | 12300 | 0 |
| 300D | 12301 | 0 |

| Line No | Assembler Message |
|---|---|
| 0 | Program assembled successfully |

(6) Write a Program to Multiply Two 8-bit Numbers. [Using **Repeated Addition** Method]

Notepad Code:

```
;Q-(6) Write a Program to Multiply Two 8-bit Numbers.
; Adding Number 1 , "Number 2 Times "  = Num1 * Num2

; Taking 3000h Location in HL Pointer
LHLD 3000h      ; H <- 3001, L <- 3000

XCHG            ; H <- D, L <- E
; Intializing the Counter
MOV C, D        ; C <- D
MVI D, 00       ; D <- 00

LXI H, 0000     ; H <- 00, L <- 00

loop: DAD D     ; HL <- HL+DE
      DCR C     ; C <- C-1

JNZ loop        ; If Zero Flag=0, goto loop

SHLD 3004h      ;  H <- 3005, L <- 3004
HLT
```

Input:

Number 1 [3000H] = 15 = (0F)H

Number 2 [3001H] = 12 = (0C)H = Counter

Output:

Output 16 Bit Answer = Register Pair Form: [3005H-3004H] = (00 B4)H = 180

**Registers**

| | | |
|---|---|---|
| *A* | 00 | |
| *BC* | 00 | 00 |
| *DE* | 00 | 0F |
| *HL* | 00 | B4 |
| *PSW* | 00 | 00 |
| *PC* | 42 | 13 |
| *SP* | FF | FF |
| *Int-Reg* | 00 | |

Data | Stack | KeyPad | **Memory** | I/O Ports

Start 3000h OK

| Address (Hex) | Address | Data |
|---|---|---|
| 3000 | 12288 | 15 |
| 3001 | 12289 | 12 |
| 3002 | 12290 | 0 |
| 3003 | 12291 | 0 |
| 3004 | 12292 | 180 |
| 3005 | 12293 | 0 |
| 3006 | 12294 | 0 |
| 3007 | 12295 | 0 |
| 3008 | 12296 | 0 |
| 3009 | 12297 | 0 |
| 300A | 12298 | 0 |
| 300B | 12299 | 0 |
| 300C | 12300 | 0 |
| 300D | 12301 | 0 |

| Line No | Assembler Message |
|---|---|
| 0 | Program assembled successfully |

<u>Input</u>:

Number 1 [3000H] = 15 = (0F)H

Number 2 [3001H] = 19 = (13)H = Counter

<u>Output</u>:

Output 16 Bit Answer = Register Pair Form: [3005H-3004H] = (01 : 29) = (01 1D)H = 285

**Registers**

| | | |
|---|---|---|
| *A* | 00 | |
| *BC* | 00 | 00 |
| *DE* | 00 | 0F |
| *HL* | 01 | 1D |
| *PSW* | 00 | 00 |
| *PC* | 42 | 13 |
| *SP* | FF | FF |
| *Int-Reg* | 00 | |

Data | Stack | KeyPad | **Memory** | I/O Ports

Start 3000h OK

| Address (Hex) | Address | Data |
|---|---|---|
| 3000 | 12288 | 15 |
| 3001 | 12289 | 19 |
| 3002 | 12290 | 0 |
| 3003 | 12291 | 0 |
| 3004 | 12292 | 29 |
| 3005 | 12293 | 1 |
| 3006 | 12294 | 0 |
| 3007 | 12295 | 0 |
| 3008 | 12296 | 0 |
| 3009 | 12297 | 0 |
| 300A | 12298 | 0 |
| 300B | 12299 | 0 |
| 300C | 12300 | 0 |
| 300D | 12301 | 0 |

| Line No | Assembler Message |
|---|---|
| 0 | Program assembled successfully |

(7) Write a Program to find Largest of two 8-bit numbers.

Notepad Code:

```
;Q-(7) Write a Program to find Largest of two 8-bit numbers.

        MVI A,00h
        LDA 3000h     ; loads value at Memory Location 3000h
        MOV B, A      ; Store it in Reg B                    [B <- Num 1]
        LDA 3001h     ; loads value at Memory Location 3001h [Acc <- Num 2]

        ; If A>B , then CF = 0 and if A<B, then CF = 1
        CMP B         ; Compare B with A

        JNC here      ; Jump if Carry flag is Reset (Carry flag = 0)
        MOV A, B      ; Num1 is Greater

here: STA 3005h       ; store result at Memory Location 3005
        HLT       ;Terminates the program
```

Registers and Memory: Input: 5 & 10 Output: 10 [Since 10>5]

Data | Stack | KeyPad | **Memory** | I/O Ports

Start 3000h OK

| Address (Hex) | Address | Data |
|---|---|---|
| 3000 | 12288 | 10 |
| 3001 | 12289 | 5 |
| 3002 | 12290 | 0 |
| 3003 | 12291 | 0 |
| 3004 | 12292 | 0 |
| 3005 | 12293 | 10 |
| 3006 | 12294 | 0 |
| 3007 | 12295 | 0 |
| 3008 | 12296 | 0 |
| 3009 | 12297 | 0 |
| 300A | 12298 | 0 |
| 300B | 12299 | 0 |
| 300C | 12300 | 0 |
| 300D | 12301 | 0 |

**Registers**

| | | |
|---|---|---|
| *A* | 0A | |
| *BC* | 0A | 00 |
| *DE* | 00 | 00 |
| *HL* | 00 | 00 |
| *PSW* | 00 | 00 |
| *PC* | 42 | 12 |
| *SP* | FF | FF |
| *Int-Reg* | 00 | |

| Line No | Assembler Message |
|---|---|
| 0 | Program assembled successfully |

SUBMITTED BY: BHAGYA VINOD RANA [***U19CS012***]