19EE412 MICROPROCESSOR AND MICROCONTROLLER LAB

EXPERIMENT NO:1

Evaluation of arithmetic expressions

Aim:

To perform 8-bit arithmetic operations such as addition, subtraction, multiplication, and division using the 8085 microprocessor.

Apparatus Required:

- Laptop with internet connection

Algorithm:

EXP.1(A)For Addition (With Carry Consideration):

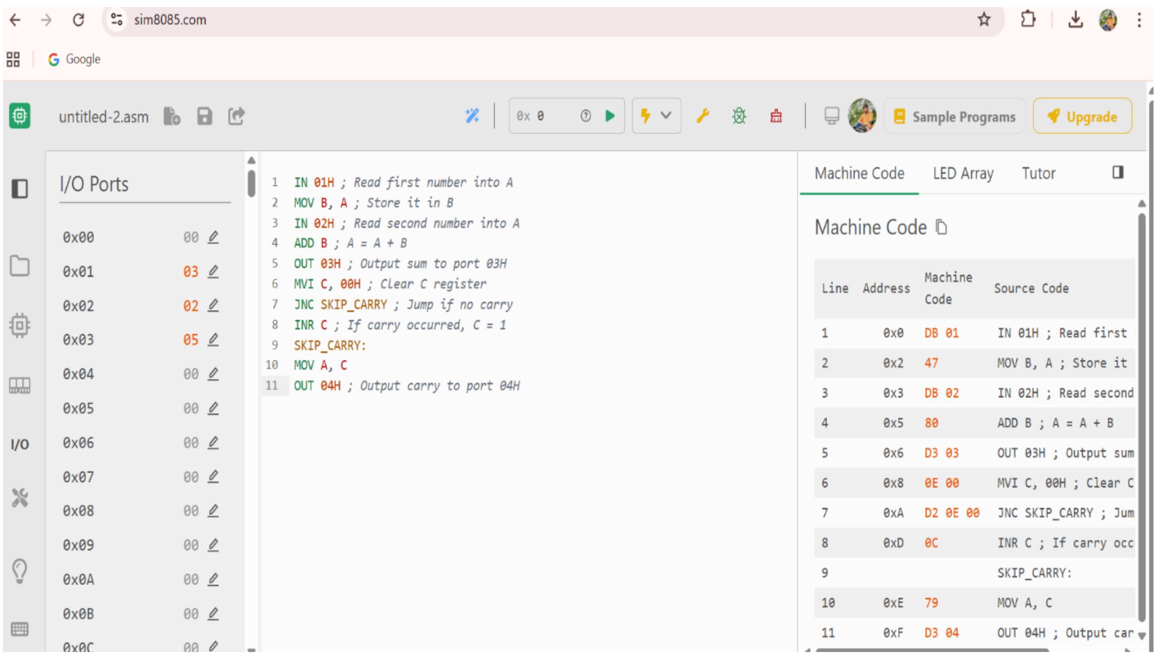
- Load the first number into register A.
- Load the second number into register B.
- Add the contents of registers A and B.
- If carry is generated, store carry in a separate location.
- Store the sum in another location.

PROGRAM:

Addition of Two 8-bit Numbers:

```
IN 01H          ; Read first number into A
MOV B, A          ; Store it in B
IN 02H          ; Read second number into A
ADD B            ; A = A + B
OUT 03H           ; Output sum to port 03H

MVI C, 00H       ; Clear C register
JNC SKIP_CARRY   ; Jump if no carry
INR C           ; If carry occurred, C = 1

SKIP_CARRY:
MOV A, C
OUT 04H           ; Output carry to port 04H
```

OUTPUT CODE:



I/O Ports

0x00	00
0x01	03
0x02	02
0x03	05
0x04	00
0x05	00
0x06	00
0x07	00
0x08	00
0x09	00
0x0A	00
0x0B	00
0x0C	00

```
IN 01H ; Read first number into A
MOV B, A ; Store it in B
IN 02H ; Read second number into A
ADD B ; A = A + B
OUT 03H ; Output sum to port 03H
MVI C, 00H ; Clear C register
JNC SKIP_CARRY ; Jump if no carry
INR C ; If carry occurred, C = 1
SKIP_CARRY:
MOV A, C
OUT 04H ; Output carry to port 04H
```

Machine Code | LED Array | Tutor

Machine Code

Line	Address	Machine Code	Source Code
1	0x0	DB 01	IN 01H ; Read first
2	0x2	47	MOV B, A ; Store it
3	0x3	DB 02	IN 02H ; Read second
4	0x5	80	ADD B ; A = A + B
5	0x6	D3 03	OUT 03H ; Output sum
6	0x8	0E 00	MVI C, 00H ; Clear C
7	0xA	D2 0E 00	JNC SKIP_CARRY ; Jum
8	0xD	0C	INR C ; If carry occ
9			SKIP_CARRY:
10	0xE	79	MOV A, C
11	0xF	D3 04	OUT 04H ; Output car

EXP.1(B) For Subtraction (Considering Greater Number):

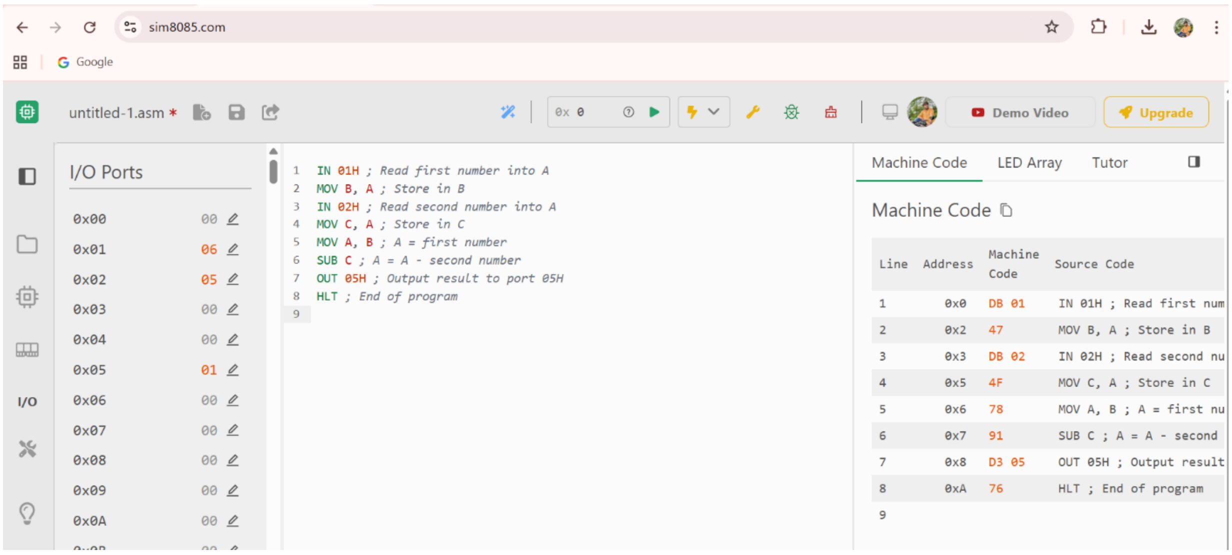
- Load the first number into register A.
- Load the second number into register B.
- Compare A and B.
- If A < B, swap the values of A and B to ensure positive result.
- Subtract the content of B from A.
- Store the result in a specified location.

PROGRAM:

Subtraction (First number - Second number)

```
IN 01H          ; Read first number into A
MOV B, A          ; Store in B
IN 02H          ; Read second number into A
MOV C, A          ; Store in C
MOV A, B        ; A = first number
SUB C          ; A = A - second number
OUT 05H         ; Output result to port 05H

HLT            ; End of program
```

OUTPUT CODE:



0x00	00
0x01	06
0x02	05
0x03	00
0x04	00
0x05	01
0x06	00
0x07	00
0x08	00
0x09	00
0x0A	00

```
IN 01H ; Read first number into A
MOV B, A ; Store in B
IN 02H ; Read second number into A
MOV C, A ; Store in C
MOV A, B ; A = first number
SUB C ; A = A - second number
OUT 05H ; Output result to port 05H
HLT ; End of program
```

Machine Code

Line	Address	Machine Code	Source Code
1	0x0	DB 01	IN 01H ; Read first num
2	0x2	47	MOV B, A ; Store in B
3	0x3	DB 02	IN 02H ; Read second nu
4	0x5	4F	MOV C, A ; Store in C
5	0x6	78	MOV A, B ; A = first nu
6	0x7	91	SUB C ; A = A - second
7	0x8	D3 05	OUT 05H ; Output result
8	0xA	76	HLT ; End of program
9			

EXP.1(C) For Multiplication:

- Load the first number into register A.
- Load the second number into register B.
- Multiply A and B using repeated addition.
- Store the result in suitable locations (including extra space if needed for higher bits).

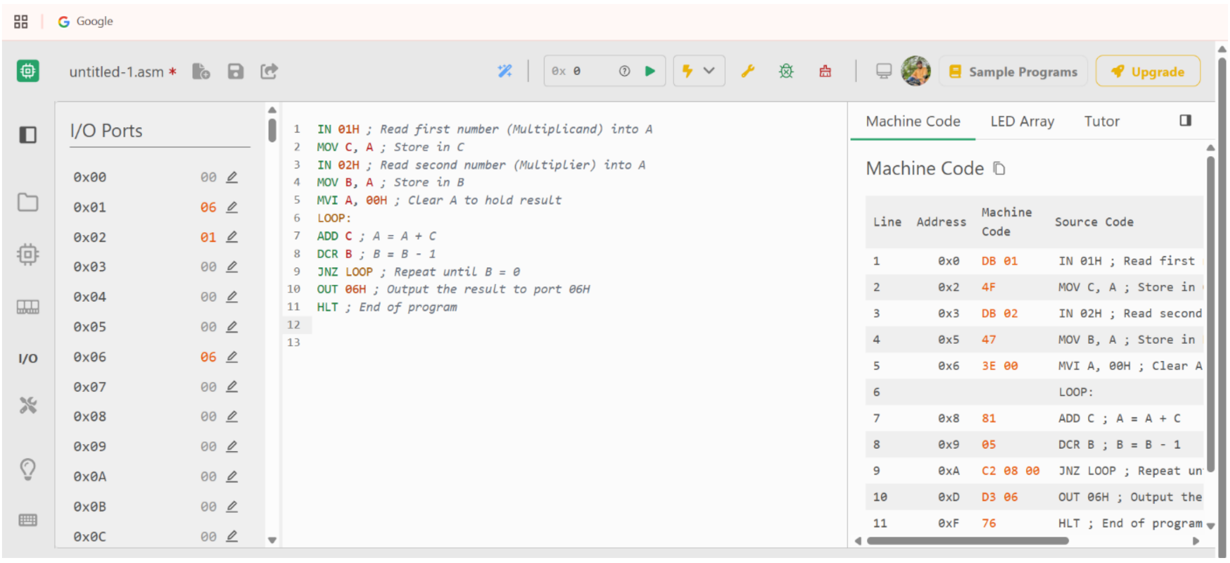
PROGRAM:

Multiplication using repeated addition:

```
IN 01H        ; Read first number (Multiplicand) into A
MOV C, A       ; Store in C

IN 02H        ; Read second number (Multiplier) into A
MOV B, A       ; Store in B

MVI A, 00H     ; Clear A to hold result
```

```
LOOP:
ADD C        ; A = A + C
DCR B        ; B = B - 1
JNZ LOOP      ; Repeat until B = 0

OUT 06H      ; Output the result to port 06H
HLT          ; End of program
```

OUTPUT CODE:



I/O Ports	
0x00	00
0x01	06
0x02	01
0x03	00
0x04	00
0x05	00
0x06	06
0x07	00
0x08	00
0x09	00
0x0A	00
0x0B	00
0x0C	00

```
IN 01H ; Read first number (Multiplicand) into A
MOV C, A ; Store in C
IN 02H ; Read second number (Multiplier) into A
MOV B, A ; Store in B
MVI A, 00H ; Clear A to hold result
LOOP:
ADD C ; A = A + C
DCR B ; B = B - 1
JNZ LOOP ; Repeat until B = 0
OUT 06H ; Output the result to port 06H
HLT ; End of program
```

Machine Code

Line	Address	Machine Code	Source Code
1	0x0	DB 01	IN 01H ; Read first
2	0x2	4F	MOV C, A ; Store in
3	0x3	DB 02	IN 02H ; Read second
4	0x5	47	MOV B, A ; Store in
5	0x6	3E 00	MVI A, 00H ; Clear A
6			LOOP:
7	0x8	81	ADD C ; A = A + C
8	0x9	05	DCR B ; B = B - 1
9	0xA	C2 08 00	JNZ LOOP ; Repeat un
10	0xD	D3 06	OUT 06H ; Output the
11	0xF	76	HLT ; End of program

EXP.1(D) For Division:

- Load the dividend into register A.
- Load the divisor into register B.
- Perform division using repeated subtraction.
- Store the quotient in one location and remainder in another.

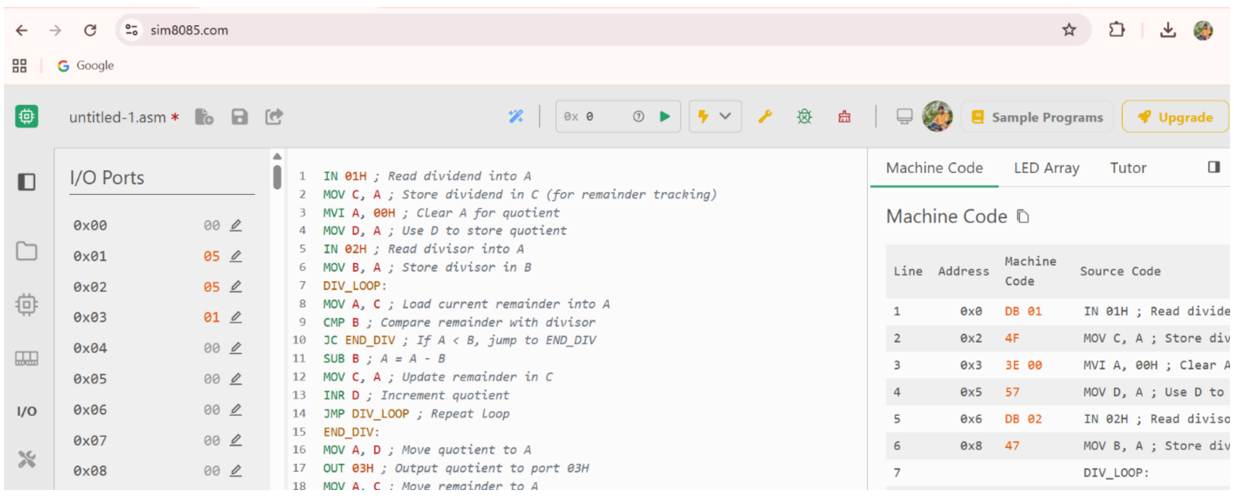
Program:

Division (Using Repeated Subtraction):

```
IN 01H        ; Read dividend into A
MOV C, A      ; Store dividend in C (for remainder tracking)
MVI A, 00H    ; Clear A for quotient
```

```
MOV D, A       ; Use D to store quotient

IN 02H         ; Read divisor into A
MOV B, A       ; Store divisor in B

DIV_LOOP:
MOV A, C       ; Load current remainder into A
CMP B          ; Compare remainder with divisor
JC END_DIV     ; If A < B, jump to END_DIV
SUB B          ; A = A - B
MOV C, A       ; Update remainder in C
INR D          ; Increment quotient
JMP DIV_LOOP   ; Repeat loop

END_DIV:
MOV A, D       ; Move quotient to A
OUT 03H        ; Output quotient to port 03H

MOV A, C       ; Move remainder to A
OUT 04H        ; Output remainder to port 04H

HLT            ; End program
```

OUTPUT PROGRAM:



I/O Ports

0x00	00
0x01	05
0x02	05
0x03	01
0x04	00
0x05	00
0x06	00
0x07	00
0x08	00

```
IN 01H ; Read dividend into A
MOV C, A ; Store dividend in C (for remainder tracking)
MVI A, 00H ; Clear A for quotient
MOV D, A ; Use D to store quotient
IN 02H ; Read divisor into A
MOV B, A ; Store divisor in B
DIV_LOOP:
MOV A, C ; Load current remainder into A
CMP B ; Compare remainder with divisor
JC END_DIV ; If A < B, jump to END_DIV
SUB B ; A = A - B
MOV C, A ; Update remainder in C
INR D ; Increment quotient
JMP DIV_LOOP ; Repeat loop
END_DIV:
MOV A, D ; Move quotient to A
OUT 03H ; Output quotient to port 03H
MOV A, C ; Move remainder to A
```

Machine Code

Line	Address	Machine Code	Source Code
1	0x0	DB 01	IN 01H ; Read divide
2	0x2	4F	MOV C, A ; Store div
3	0x3	3E 00	MVI A, 00H ; Clear A
4	0x5	57	MOV D, A ; Use D to
5	0x6	DB 02	IN 02H ; Read diviso
6	0x8	47	MOV B, A ; Store div
7			DIV_LOOP:

Result:

The 8-bit arithmetic operations using the 8085 microprocessor have been successfully executed and verified using memory access for input and output.