Assembly Language Programming :-

ALP - Type of low-level programming language that communicate directly with computer hardware

- A program is a set of instructions arranged in the specific sequence to do the specific task.
- The process of writing the set of instructions which tells the Microprocessor what to do is called Programming.

Or

Programming is the process of telling the processor exactly how to solve a problem.

- To do this, the programmer must speak to the processor in a language which processor can understand.
- Steps in Programming:

(1) Find out which task is to be performed.

(2) Design the problem solution

(3) Once the program is specified & designed, it can be implemented Implementation begins with the the coding of the program.

(4) Once the program is coded, the next step is debugging.

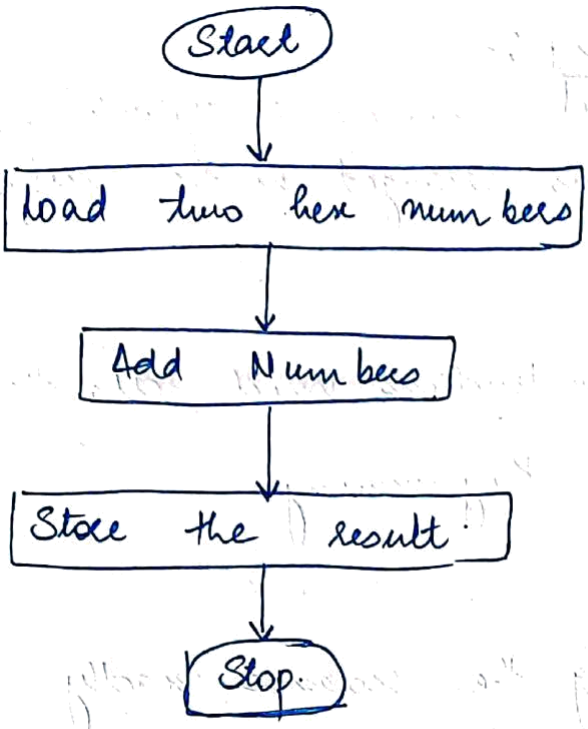
Eg WA ALP to add two numbers.

The three task are involved in the program

→ Load two hex numbers.

→ Add numbers

→ Store the result in memory.



Task 1

MVI A, 20H	Load 20H as first number in register A.
MVI B, 40H	Load 40H as second number in register B.

Task 2

ADD B	Add two numbers and save results in register A.

Task 3

STA 2200H	Store the result in memory location 2000H.
HLT	Stop the program execution.

Exp 1. Store 8-bit data in memory.

Input: 2000 H = 52 H

(P1)

```
MVI A, 52H    ; Store 52H in the accumulator.
STA 2000H     ; Copy accumulator content at 2000 H.
HLT           ; Terminate Program Execution
```

for Simulator

Memory location	Mnemonics	Opcode.
2000	MVI A, 52 H	3E
2001		52
2002	STA 2000 H	32
2003		00
2004		20
2005	HLT	76

(P2)

```
LXI H, 2000H  ; Load HL with 2000H.
MVI M, 52H    ; Store 52H in memory location pointed
                by HL register pair.
HLT           ; Terminate Program Execution
```

* In P1, direct addressing mode instruction is used.
In P2, indirect addressing mode instruction is used.

Exp 2. Exchange the content of memory location.

(P1)

```
LDA 1000H ; Get the content of memory location 1000H
            into the accumulator.
MOV B, A  ; Save the content in B register.
LDA 2000H ; Get the content of memory location 2000H
            into the accumulator.
STA 1000H ; Store the content of accumulator at
            address 1000H.
MOV A, B  ; Get the same content back to register A.
STA 2000H ; Store the content of accumulator at
            address 2000H.
HLT       ; Terminate program execution.
```

2000	3A	LDA 1000H
2001	00	
2002	20	
2003	47	MOV B,A
2004	3A	LDA 2000H
2005	00	
2006	20	
2007	32	STA 2000H
2008	00	
2009	20	
200A	78	MOV A,B
200B	32	STA 2000H
200C	00	
200D	20	
200E	76	HLT

(P2)

```
LXI H, 1000H ; Initialize HL register pair as Memory location
               1000H.
LXI D, 2000H ; Initialize DE register pair as memory location
               2000H.
MOV B, M     ; Get the content of Memory location 1000H
               into B register.
LDAX D       ; Get the content of Memory location 2000H
               into A register.
MOV M, A     ; Store the content of A register into Memory
               location 1000H.
MOV A, B     ; Copy the content of B register into
               accumulator.
STAX D       ; Store the content of A register into
               Memory location 2000H.
HLT          ; Terminate Program Execution.
```

Exp 3. Add two 8- bit numbers.

Input 2000 H = 14 H
2001 H = 89 H

Output 2002 H = 14H + 89H
= 9D H

(P)

Memory location	Mnemonics	Opcode	Description
2500	LXI H, 2000H	21	HL points 2000H
2501		00	
2502		20	
2503	MOV A, M	7E	Get first operand
2504	INX H	23	HL points 2001 H
2505	ADD M	86	Add second Operand
2506	INX H	23	HL points 2002 H
2507	MOV M, A	77	Store result at 2002 H
2508	HLT	76	Terminate Program Execution

Exp 4. Subtract two 8- bit numbers.

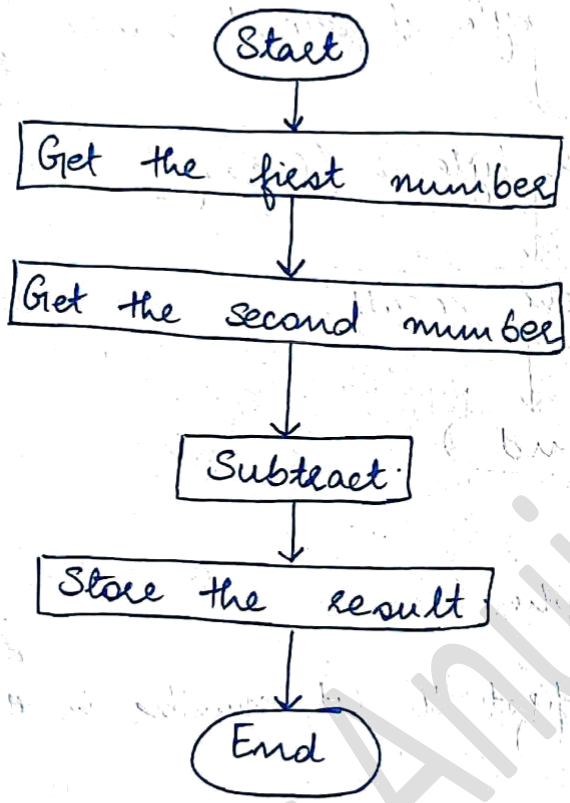
Input : 2000 H = 51 H
2001 H = 19 H

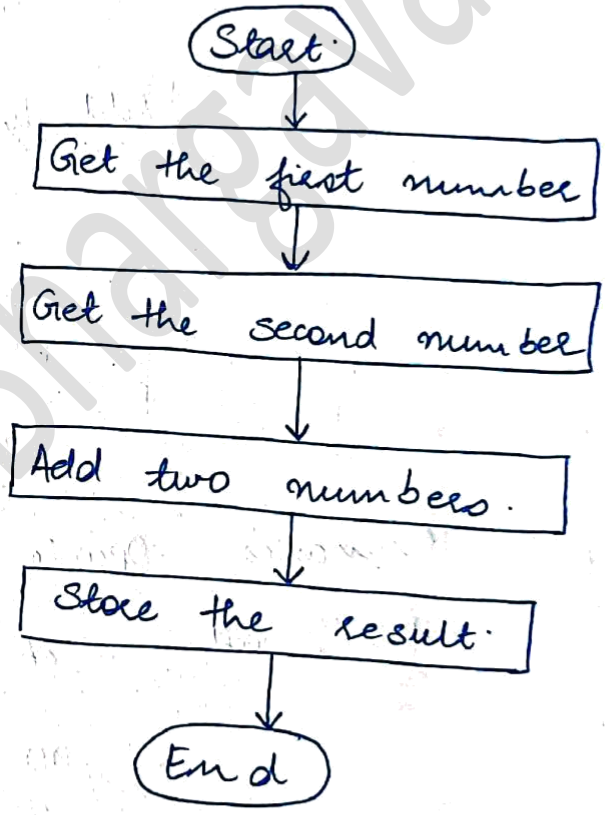
Output : 2002H = 51 H - 19H
= 38H

(P)

2500	LXI H, 2000H	21	HL points 2000 H
2501		00	
2502		20	
2503	MOV A., M	7E	Get first operand

2504	INX H	23	HL points 2001 H
2505	SUB M	96	Subtract Second Operand
2506	INX H	23	HL points 2002 H
~~2507~~	~~SBB M~~		~~Subtract second operand~~
2507	MOV M,A	77	Store results at 2002-H
2508	HLT	76	Terminate Program





Exp 4. Add two 16-bit numbers.

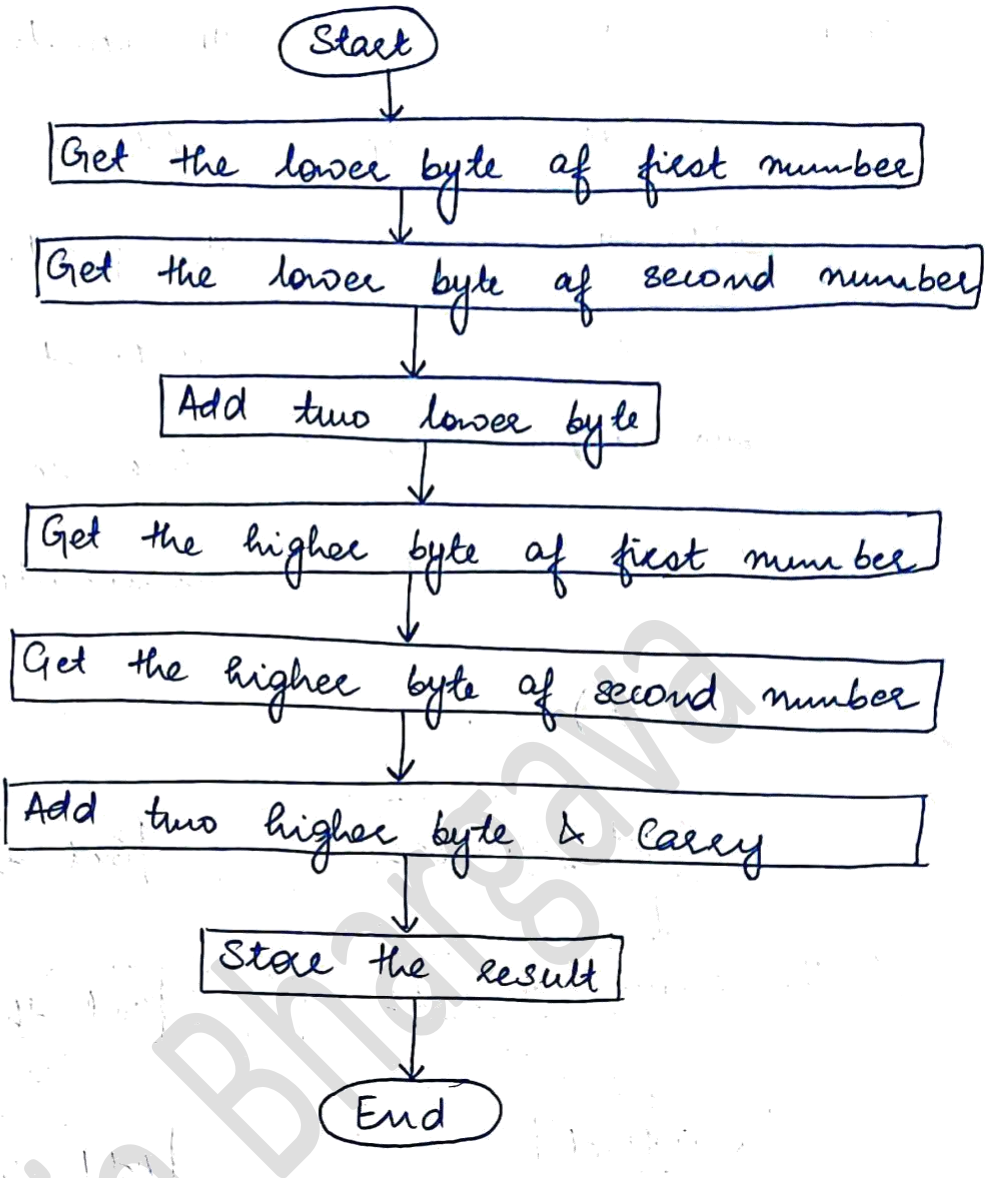
Input: 1 C 1 5 H → 0001 1100 0001 0101

5 A B 7 H → + 0101 1010 1011 0111

Output: 7 6 C C H 0111 0110 1100 1100

7 6 12 12

Input	2000	15 H
	2001	1C H
	2002	B7 H
	2003	5A H
Output	2004	CC H
	2005	76 H



(P1)

Memory L.	Mnemonics	Opcode	Description
2500	LHLD 2000H	2A	Get first 16-bit number in HL.
2501		00	
2502		20	
2503	XCHG	EB	Save first 16-bit number in DE.
2504	LHLD 2002H	2A	Get second 16-bit number in HL.
2505		02	
2506		20	
2507	DAD D	19	Add DE & HL.

2508	SHLD 2004 H	22	Store 16 bit result in 2004 & (4)
2509		04	2005 H
250A		20	
250B	HLT	76	Terminate Program Execution

(P2)

2500	LHLD 2000H	2A	Get first 16-bit number in HL.
2501		00	
2502		20	
2503	XCHG	EB	Save first 16-bit number in DE.
2504	LHLD 2002 H	2A	Get second 16-bit number in HL
2505		02	
2506		20	
2507	MOV A, E	7B	Get lower byte of the first number
2508	ADD L	85	Add lower byte of the first number
2509	MOV L, A	6F	Store result in L register.
250A	MOV A, D	7A	Get higher byte of the first number.
250B	ADC H	8C	Add higher byte of second no. with carry
250C	MOV H, A	67	Store result in H-register.
250D	SHLD 2004H	22	Store result in 2004 & 2005 H.
250E		04	
250F		20	
2510	HLT	76	Terminate Program Execution

Exp 5. Subtract two 16-bit numbers.

Input 6A 19 H → 0110 1010 0001 1001

5C 15 H → 0101 1100 0001 0101

Output 0E04 H 0000 1110 0000 0100

0 E 0 4

Input 2000H 19 H

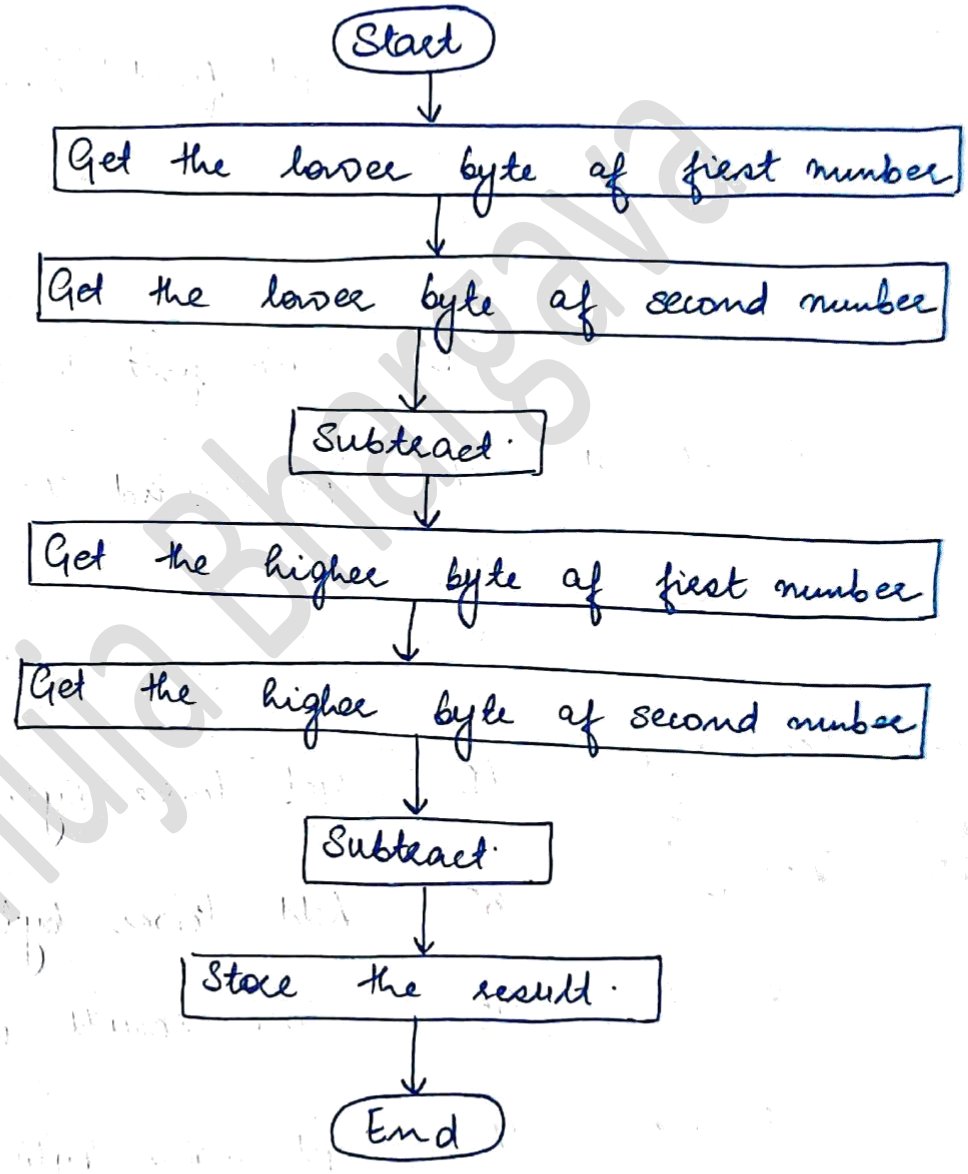
2001 H 6A H

2002 H 15 H

2003 H 5C H

Output 2004 H 04 H

2005 H 0E H



```
LHLD 2000 H ; Get first 16-bit number in HL.
XCHG        ; Save first number in DE.
LHLD 2002 H ; Get second 16-bit number in HL.
MOV A, E    ; Get lower byte of first number.
SUB L       ; Subtract lower byte of second number.
MOV L, A    ; Store result in L register.
MOV A, D    ; Get higher byte of first number.
SBB H       ; Subtract higher byte of second no. with borrow
MOV H, A    ; Store 16-bit result in memory location
            ; 2004 H & 2005 H.
SHLD 2004 H ;
HLT         Terminate Program Execution.
```

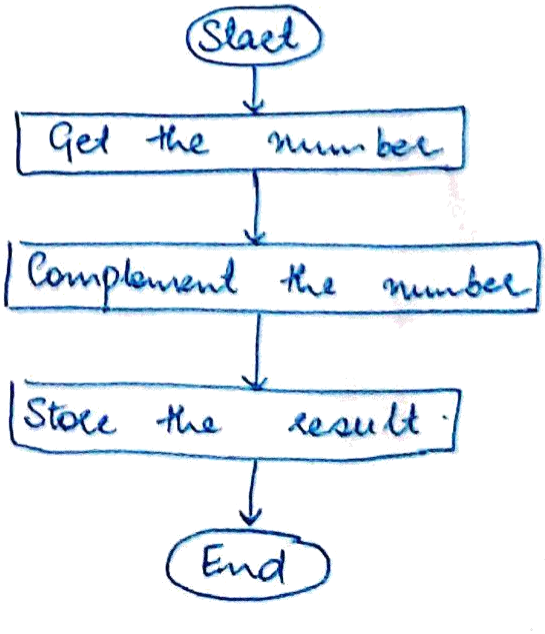
Exp 6. Find 1's complement of a number.

Input: 2200 H 55 H

Output: 2300 H AA H

(P)

```
LDA 2200H   Get the number
CMA         Complement number
STA 2300H   Store the result
HLT         Terminate Program Execution
```



Exp 7. Find the 2's complement of a number.

Input 2200 H 55 H

Output 2300 H AB H

(P)

```
LDA 2200H    Get the number
CMA          Complement the number
ADI, 01 H    Add one in the number
STA 2300 H   Store the result
HLT          Terminate program execution
```