

Unit -IV

Programming of 8085 Microprocessor

Basic Programs for practice:

Programs:

1. Place 05H data in register B.

Program	Comments
MVI B,05H	Move immediate data 05H into reg. B
HLT	Stop

Result:

Register B = 05H

2. Get 95H data in register C and move this data into register A.

Program	Comments
MVI C,95H	Move immediate data 95H into reg. C
MOV A,C	Move data from reg. C to reg. A
HLT	Stop

Result:

Register A = 95H

3. Move the content of memory location 2000H into register E. The contents of memory location 2000H is 76H.

Program	Comments
LXI H,2000H	Load/Get 2000H memory addr. in HL pair
MOV E,M	Move the contents of memory, whose address is in HL pair into reg. E
HLT	Stop

Result:

Register E = 76H

4. Place the contents of memory location 2000H into register B and that of 2001H into register C. The contents of memory location 2000H and 2001H 35H and 25H respectively.

Program	Comments
LXI H,2000H	Load/Get 2000H memory address in HL pair
MOV B,M	Move the contents of memory location 2000H into reg. B
INX H	Increment HL pair by 1 i.e. get 2001H memory addr. in HL pair
MOV C,M	Move the contents of memory location 2001H into reg. C
HLT	Stop

Result:

Register B = 35H

Register C = 25H

Q .5. Take F000H memory address in HL pair and move the data from memory to reg, C and increment it by 1. Let the data at F000H be 49H.

Program	Comments
LXI H, F000H	Load/Get F000H memory address in HL pair
MOV C,M	Move the contents of memory location 2000H into reg. C
INR C	Increment the contents of reg. C by 1
HLT	Stop

Result:

REG. C=49H +01= 4AH

Q.6. Subtract the contents of reg. B from reg. A. Load 3000H data into reg A and 3001H data into reg. B. Let the data of 3000H and 3001H be 45H and 33H. What will be the result.

Program	Comments
LXI H, 3000H	Load/Get 3000H memory address in HL pair
MOV A,M	Move the contents of memory location 2000H into reg. C
INX H	Increment the contents of reg. HL pair by 1
MOV B,M	Move the contents of memory location 3001H into reg. B
SUB B	Subtracts the contents of reg. B from reg. A
HLT	Stop

Result=Reg. A=12H

45

- 33

12H

Assembly Language Programming of 8085 Microprocessor:

- 1. Write an ALP (Assembly Language Program) to add two 8-bit numbers, whose sum is also 8-bit.**

→ Let us assume:

First 8-bit number at 2000H = 33H

Second 8-bit number at 2001H = 55H

Result at 2002H = 88H

Program	Comments
LXI H,2000H	Load/Get addr. of first number 2000H into HL pair
MOV A,M	Move the first number from memory to Accumulator i.e. reg.A
INX H	Get addr. of second number 2001H into HL pair
ADD M	Add first number with second number i.e. $[A] \leftarrow [A] + [[HL]]$
STA 2002H	Store result from Accumulator to 2002H
HLT	Stop

- 2. Write an ALP (Assembly Language Program) to perform subtraction of two 8-bit numbers.**

→ Let us assume:

First 8-bit number at 2000H = 97H

Second 8-bit number at 2001H = 43H

Result at 2002H = 54H

Program	Comments
LXI H,2000H	Load/Get addr. of first number 2000H into HL pair
MOV A,M	Move the first number from memory to Accumulator i.e. reg. A
INX H	Get addr. of second number 2001H into HL pair
SUB M	Subtract second number from first number i.e. $[A] \leftarrow [A] - [[HL]]$
STA 2002H	Store result from Accumulator to 2002H
HLT	Stop

3. Write an ALP (Assembly Language Program) to logically AND two 8-bit numbers.

→ Let us assume:

First 8-bit number at 2000H = 97H=1001 0111

Second 8-bit number at 2001H = 43H=0100 0011

Result at 2002H = 03H=0000 0011

Program	Comments
LXI H,2000H	Load/Get addr. of first number 2000H into HL pair
MOV A,M	Move the first number from memory to Accumulator i.e. reg. A
INX H	Get addr. of second number 2001H into HL pair
ANA M	Logically AND first number with second number i.e. $[A] \leftarrow [A] \wedge [[HL]]$
STA 2002H	Store result from Accumulator to 2002H
HLT	Stop

4. Write an ALP (Assembly Language Program) to logically OR two 8-bit numbers.

→ Let us assume:

First 8-bit number at 2000H = 66H=0110 0110

Second 8-bit number at 2001H = 45H=0100 0101

Result at 2002H = 67H=0110 0111=67H

Program	Comments
LXI H,2000H	Load/Get addr. of first number 2000H into HL pair
MOV A,M	Move the first number from memory to Accumulator i.e. reg. A
INX H	Get addr. of second number 2001H into HL pair
ORA M	Logically OR first number with second number i.e. $[A] \leftarrow [A] \vee [[HL]]$
STA 2002H	Store result from Accumulator to 2002H
HLT	Stop

5. Write an ALP (Assembly Language Program) to logically EX-OR two 8-bit numbers.

→ Let us assume:

First 8-bit number at 2000H = 78H=0111 1000

Second 8-bit number at 2001H = 32H=0011 0010

Result at 2002H = 4AH=0100 1010=4AH

Program	Comments
LXI H,2000H	Load/Get addr. of first number 2000H into HL pair
MOV A,M	Move the first number from memory to Accumulator i.e. reg. A
INX H	Get addr. of second number 2001H into HL pair
XRA M	Logically Ex-OR first number with second number i.e. $[A] \leftarrow [A] \oplus [[HL]]$
STA 2002H	Store result from Accumulator to 2002H
HLT	Stop

6. Write an ALP (Assembly Language Program) to find 1's complement of an 8-bit number.

→ Let us assume:

8-bit number at 2000H = 97H

Binary of number = 1001 0111

1's complement of number = 0110 1000

Result at 2001H = 68H

Program	Comments
LXI H,2000H	Load/Get addr. of number 2000H into HL pair
MOV A,M	Move the number from memory to Accumulator i.e. reg. A
CMA	Take 1's complement of the contents of Accumulator i.e. $[A] \leftarrow [\bar{A}]$
STA 2001H	Store result from Accumulator to 2001H
HLT	Stop

7. Write an ALP (Assembly Language Program) to find 2's complement of an 8-bit number.

→ Let us assume:

8-bit number at 2000H = 97H

8421 8421

Binary of number = 1001 0111

1's complement of number = 0110 1000

Add 1 in 1's complement of no. = 1

_-----

8421 8421

0110 1001=69H

Result at 2001H = 69H

Program	Comments
LXI H,2000H	Load/Get addr. of number 2000H into HL pair
MOV A,M	Move the number from memory to Accumulator i.e. reg. A
CMA	Take 1's complement of the contents of Accumulator I.e. $[A] \leftarrow [\bar{A}]$
ADI 01H/INR A	Add 1 in 1's complement of number in Accumulator=2's complement of number
STA 2001H	Store result from Accumulator to 2001H
HLT	Stop

8. Write an ALP (Assembly Language Program) to shift/rotate an 8-bit number by 1-bit in left direction.

→ Let us assume:

8-bit number at 2000H = 48H

Binary of number = 0100 1000

Shift/Rotate number by 1-bit in left direction = 1001 0000

Result at 2001H = 90H

Program	Comments
LXI H,2000H	Load/Get addr. of number 2000H into HL pair
MOV A,M	Move the number from memory to Accumulator i.e. reg. A
RLC	Shift/Rotate the contents of Accumulator in left direction by 1-bit Without carry
STA 2001H	Store result from Accumulator to 2001H
HLT	Stop

9. Write an ALP (Assembly Language Program) to shift/rotate an 8-bit number by 1-bit in right direction.

→ Let us assume:

8-bit number at 2000H = 48H

Binary of number = 0100 1000

Shift/Rotate number by = 0010 0100

1-bit in left direction

Result at 2001H = 24H

Program	Comments
LXI H,2000H	Load/Get addr. of number 2000H into

	HL pair
MOV A,M	Move the number from memory to Accumulator i.e. reg. A
RRC	Shift/Rotate the contents of Accumulator in right direction by 1-bit Without carry
STA 2001H	Store result from Accumulator to 2001H
HLT	Stop

10. Write an ALP (Assembly Language Program) to shift/rotate an 8-bit number by 2-bit in left direction.

→ Let us assume:

8-bit number at 2000H = 48H

Binary of number = 0100 1000

Shift/Rotate number by 1-bit in left direction = 1001 0000 =90

Shift/Rotate number by 1-bit in left direction = 0010 0001=21

Result at 2001H = 21H

Program	Comments
LXI H,2000H	Load/Get addr. of number 2000H into HL pair
MOV A,M	Move the number from memory to Accumulator i.e. reg. A
RLC	Shift/Rotate the contents of Accumulator in left direction by 1-bit
RLC	Shift/Rotate the contents of Accumulator in left direction by 1-bit
STA 2001H	Store result from Accumulator to 2001H
HLT	Stop

11. Write an ALP (Assembly Language Program) to shift/rotate an 8-bit number by 2-bit in right direction.

→ Let us assume:

8-bit number at 2000H = 48H

Binary of number = 0100 1000

Shift/Rotate number by 1-bit in left direction = 0010 0100

Shift/Rotate number by 1-bit in left direction = 0001 0010

Result at 2001H = 12H

Program	Comments
LXI H,2000H	Load/Get addr. of number 2000H into HL pair
MOV A,M	Move the number from memory to Accumulator i.e. reg. A
RRC	Shift/Rotate the contents of Accumulator in right direction by 1-bit
RRC	Shift/Rotate the contents of Accumulator in right direction by 1-bit
STA 2001H	Store result from Accumulator to 2001H
HLT	Stop

12. Write an ALP (Assembly Language Program) to find larger number between two 8-bit numbers.

→ Let us assume:

First 8-bit number at 2000H = 97H=97H

Second 8-bit number at 2001H = 43H=43H

Result at 2002H = 97H= 54H

D D

first no. 43H

second no. 97H

- -1

-1 46H

From Rohini Birajdar to Everyone: 03:58 PM

yes

From Gavhane Arti to Me: (Direct Message) 03:58 PM

yes

From V i k a s G a v a l i to Me: (Direct Message) 03:58 PM

good morning mam

From Sakshi Bandichawade to Me: (Direct Message) 03:58 PM

s mam

From V i k a s G a v a l i to Me: (Direct Message) 03:58 PM

yes mam

From Sumit '-' Bhosle to Everyone: 03:58 PM

Yes mam

From V i k a s G a v a l i to Me: (Direct Message) 03:58 PM

yes

From Mohit Ashte... to Everyone: 03:58 PM

good morning mam

From Vijay Shinde to Everyone: 03:59 PM

yes

From Sanket Admane to Everyone: 03:59 PM

s

From Rohini Birajdar to Me: (Direct Message) 03:59 PM

yes mam

From V i k a s G a v a l i to Me: (Direct Message) 04:00 PM

yes

yes mam

From Yadav saraswati to Everyone: 04:00 PM

no

From Jayshri Panchal to Everyone: 04:00 PM

yes

From Sakshi Bandichawade to Me: (Direct Message) 04:00 PM

s

From Mohit Ashte... to Everyone: 04:00 PM

mam opcode 8085 app madhay type your code hear manat ahay mam

From Yadav saraswati to Me: (Direct Message) 04:00 PM

mam repeat

From Parth Joshi to Everyone: 04:01 PM

konta use karaycha mam

From Mohit Ashte... to Everyone: 04:01 PM

mobile ver ma

yes

From Gavhane Arti to Me: (Direct Message) 04:01 PM

nahi

From Mohit Ashte... to Everyone: 04:02 PM

ok

mam app name sanga

From V i k a s G a v a l i to Me: (Direct Message) 04:02 PM

yes

From Mohit Ashte... to Everyone: 04:02 PM

ok mam

From Shubhangi biradar to Me: (Direct Message) 04:05 PM

54h

From Rohan Rakh-_*_ to Me: (Direct Message) 04:06 PM

+

From Balasaheb_karad to Me: (Direct Message) 04:06 PM

positive

From Shubhangi biradar to Me: (Direct Message) 04:06 PM

positive

From Jadhav Sadhana to Me: (Direct Message) 04:06 PM

+ve

From Balasaheb_karad to Me: (Direct Message) 04:07 PM

-54

From Rohini Birajdar to Me: (Direct Message) 04:08 PM

6

From Shubhangi biradar to Me: (Direct Message) 04:08 PM

6

From Vaibhav hipparge to Me: (Direct Message) 04:08 PM

6

From Kshitija Ganage to Me: (Direct Message) 04:08 PM

4

From Shubhangi biradar to Me: (Direct Message) 04:08 PM

4

From Sumit '-' Bhosle to Me: (Direct Message) 04:08 PM

4

From Rohini Birajdar to Me: (Direct Message) 04:08 PM

4

From Mohit Ashte... to Everyone: 04:08 PM

4

From Rohan Rakh-_*_ to Me: (Direct Message) 04:08 PM

4

From Shaikh Wasim to Me: (Direct Message) 04:08 PM

4

From Shubhangi biradar to Me: (Direct Message) 04:09 PM

negative result

From Rohan Rakh-_*_ to Me: (Direct Message) 04:10 PM

s

From Kshitija Ganage to Me: (Direct Message) 04:10 PM

yes

From Shubhangi biradar to Me: (Direct Message) 04:10 PM

s mam

From pallavi giri to Everyone: 04:10 PM

yes

From Sanket Jangale to Everyone: 04:10 PM

s

From Rohini Birajdar to Me: (Direct Message) 04:13 PM

yes

From Jadhav Sadhana to Me: (Direct Message) 04:13 PM

s

From Amit Biradar to Me: (Direct Message) 04:16 PM

STA 2002H I think

From Jadhav Sadhana to Me: (Direct Message) 04:16 PM

STA2002H

From Amit Biradar to Me: (Direct Message) 04:18 PM

0007H remove karun бага mam ekda

From Sakshi Bandichawade to Me: (Direct Message) 04:19 PM

Go use kara na

ekda

From Amit Biradar to Me: (Direct Message) 04:19 PM

ok mam

From Sakshi Bandichawade to Me: (Direct Message) 04:19 PM

ok

From V i k a s G a v a l i to Me: (Direct Message) 04:21 PM

yes

From Kshitija Ganage to Me: (Direct Message) 04:23 PM

no

From Sakshi Bandichawade to Me: (Direct Message) 04:23 PM

no

From V i k a s G a v a l i to Me: (Direct Message) 04:23 PM

yes mam

From Amit Biradar to Me: (Direct Message) 04:23 PM

no

From Vaibhav hipparge to Everyone: 04:23 PM

No

From Jadhav Sadhana to Me: (Direct Message) 04:23 PM

no

From Mohit Ashte... to Everyone: 04:23 PM

no mam

From V i k a s G a v a l i to Me: (Direct Message) 04:23 PM

haa

From Aditya Belurkar to Me: (Direct Message) 04:29 PM

screen share kara

From Arvind Biradar to Me: (Direct Message) 04:29 PM

screen share karo mam

ok mam

From Datta Shrimale to Me: (Direct Message) 04:36 PM

mam address dusara gheun бага

From Rohini Birajdar to Me: (Direct Message) 04:37 PM

opcode 8085 in mobile

From Datta Shrimale to Me: (Direct Message) 04:37 PM

ok

From Gs to Me: (Direct Message) 04:37 PM

opcode 8085 for android

From Safura Patel to Me: (Direct Message) 04:37 PM

Urcode 8085 software for microprocessor

<https://play.google.com/store/apps/details?id=balti.opcode8085> for mobile

From Rohini Birajdar to Me: (Direct Message) 04:37 PM

<https://play.google.com/store/apps/details?id=balti.opcode8085>

Program	Comments
LXI H,2000H	Load/Get addr. of first number 2000H into HL pair

MOV A,M	Move the first number from memory to Accumulator i.e. reg. A
INX H	Get addr. of second number 2001H into HL pair
CMP M	Compare first number with second number i.e. [A] - [[HL]]
JNC Go	Jump if no carry to label Go, as the larger number is in Accumulator
MOV A,M	If carry then larger number is in memory move to Accumulator
GO: STA 2002H	Store larger number from Accumulator to 2002H
HLT	Stop

13. Write an ALP (Assembly Language Program) to find smaller number

between two 8-bit numbers.

→ Let us assume:

First 8-bit number at 2000H = 44H

Second 8-bit number at 2001H = 78H

Result at 2002H = 44H

Program	Comments
LXI H,2000H	Load/Get addr. of first number 2000H into HL pair
MOV A,M	Move the first number from memory to Accumulator i.e. reg. A
INX H	Get addr. of second number 2001H into HL pair
CMP M	Compare first number with second number i.e. [A] - [[HL]]
JC Go	Jump if no carry to label Go, as the

	smaller number is in Accumulator
MOV A,M	If carry then smaller number is in memory move to Accumulator
GO: STA 2002H	Store larger number from Accumulator to 2002H
HLT	Stop

14. Write an ALP (Assembly Language Program) to add two 8-bit numbers, whose sum is more than 8-bits.

→ Let us assume: **11**

First 8-bit number at 2000H = 98H

Second 8-bit number at 2001H = 9AH

+ -----

01 3 2

MSB LSB (result)

Result at **2002H =32H**

and **2003H = 01H**

Program	Comments
MVI C,00H	Initialize register C for carry
LXI H,2000H	Load/Get addr. of first number 2000H into HL pair
MOV A,M	Move the first number from memory to Accumulator i.e. reg. A
INX H	Get addr. of second number 2001H into HL pair
ADD M	Add first number with second number i.e. $[A] \leftarrow [A] + [[HL]]$
JNC GO	Jump if no carry to the label go
INR C	If carry then increment register C by 1
GO: STA 2002H	Store LSB result from Accumulator to 2002H
MOV A,C	Move carry(MSB result) from reg. C to Accumulator
STA 2003H	Store MSB result from Accumulator to 2003H
HLT	Stop

15. Write an ALP (Assembly Language Program) to add two 16-bit numbers, whose sum is also 16-bit.

→ Let us assume:

First 16-bit number at 2000H = 33H = 4533H

and at 2001H = 45H

Second 16-bit number at 2002H = 21H = 1121H

and at 2003H = 11H

Result at 2004H = 54H = +5654H

and at 2005H = 56H

Program	Comments
LHLD 2000H	Load/Get first 16-bit number in HL pair from 2000H and 2001H
XCHG	Exchange HL pair with DE pair i.e. get first 16-bit number in DE pair
LHLD 2002H	Load/Get second 16-bit number in HL pair from 2002H and 2003H
DAD D	Add first number with second number i.e. $[HL] \leftarrow [HL] + [DE]$
SHLD 2004H	Store result from HL pair to 2004H and 2005H
HLT	Stop

16. Write an ALP (Assembly Language Program) to add two 16-bit numbers, whose sum is more than 16-bit.

→ Let us assume: 11

First 16-bit number at 2000H = 45H= A645H

and at 2001H =A6H

Second 16-bit number at 2002H = 23H= 9B23H

and at 2003H =9BH

+

Result at (LSB) 2004H = 68H= 01 4 16 8

and (LSB) at 2005H =41H

and (MSB) at 2006H =01H

Program	Comments
MVI C,00H	Initialize register C for carry
LHLD 2000H	Load/Get first 16-bit number in HL pair from 2000H and 2001H
XCHG	Exchange HL pair with DE pair i.e. get first 16-bit number in DE pair
LHLD 2002H	Load/Get second 16-bit number in HL pair from 2002H and 2003H
DAD D	Add first number with second number i.e. $[HL] \leftarrow [HL] + [DE]$
JNC GO	Jump if no carry to label GO
INR C	If carry increment register C by 1
GO: SHLD 2004H	Store LSB result from HL pair to 2004H and 2005H
MOV A,C	Move carry (MSB result) from reg. C to Accumulator
STA 2006H	Store MSB result (Carry) from Accumulator to 2006H
HLT	Stop

17. Write an ALP (Assembly Language Program) to find 1's complement of an 16-bit number.

→ Let us assume:

16-bit LSB number at 2000H = 97H = 6597H

and at MSB 2001H =65

6 5 9 7

8421 8421 8421 8421

Binary of number = 0110 0101 1001 0111

1's complement of number =1001 1010 0110 1000

9 A 6 8H

Result at LSB 2002H = 68H

and at MSB 2003 =9AH

Program	Comments
LXI H,2000H	Load/Get addr. of LSB number 2000H into HL pair
MOV A,M	Move the LSB number from memory to Accumulator i.e. reg. A
CMA	Take 1's complement of the contents of Accumulator i.e. LSB number [A] ← \bar{A}
STA 2002H	Store LSB result from Accumulator to 2002H
INX H	Load/Get addr. of MSB number 2001H into HL pair
MOV A,M	Move the MSB number from memory to Accumulator i.e. reg. A
CMA	Take 1's complement of the contents of Accumulator i.e. MSB number [A] ← \bar{A}
STA 2003H	Store MSB result from Accumulator to 2003H
HLT	Stop

result 2's complement Result in binary = 1010 0100 0111 0100

A 4 7 4 H

LSB Result at 2002H = 74H

and MSB result at 2003H =A4H

Program	Comments
MVI C,00H	Initialize register C for carry
LXI H,2000H	Load/Get addr. of LSB number 2000H into HL pair
MOV A,M	Move the LSB number from memory to Accumulator i.e. reg. A
CMA	Take 1's complement of the contents of Accumulator i.e. LSB number I.e. $[A] \leftarrow [\bar{A}]$
ADI 01H/INR A	Add 1 in 1's complement of number in Accumulator=2's complement of LSB number
STA 2002H	Store LSB result from Accumulator to 2002H
JNC GO	Jump if no carry to the label GO
INR C	If carry then increment register C by 1
GO: INX H	Load/Get addr. of MSB number 2001H into HL pair
MOV A,M	Move the MSB number from memory to Accumulator i.e. reg. A
CMA	Take 1's complement of the contents of Accumulator i.e. MSB number I.e. $[A] \leftarrow [\bar{A}]$
ADD C	Add carry to MSB result
STA 2003H	Store MSB result from Accumulator to 2003H
HLT	Stop

19. Write an ALP (Assembly Language Program) to find larger number between a series of 8-bit numbers.

→ Let us assume:

Count of series at 2000H =04H

First 8-bit number at 2001H = 77H

Second 8-bit number at 2002H = 43H

Third 8-bit number at 2003H = 67H

Fourth 8-bit number at 2004H = 89H

Result at 2005H =89H

Program	Comments
LXI H,2000H	Load/Get addr. of count 2000H into HL pair
MOV C,M	Move the count from memory to Accumulator i.e. reg. C
INX H	Get addr. of first number 2001H into HL pair
MOV A,M	Move the first number from memory to Accumulator i.e. reg. A
DCR C	Decrement count by 1
LOOP: INX H	Load/Get addr. of next number into HL pair
CMP M	Compare previous number with next number i.e. [A] - [[HL]]
JNC Go	Jump if no carry to label Go, as the larger number is in Accumulator
MOV A,M	If carry then larger number is in

	memory move to Accumulator
GO: DCR C	Decrement count by 1
JNZ LOOP	Jump if count is not zero
GO: STA 2005H	Store larger number from Accumulator to 2005H
HLT	Stop

20. Write an ALP (Assembly Language Program) to find smaller number

between a series of 8-bit numbers.

→ Let us assume:

Count of series at 2000H =04H

First 8-bit number at 2001H = 77H

Second 8-bit number at 2002H = 43H

Third 8-bit number at 2003H = 67H

Fourth 8-bit number at 2004H = 89H

Result at 2005H = 43H

Program	Comments
LXI H,2000H	Load/Get addr. of count 2000H into HL pair
MOV C,M	Move the count from memory to Accumulator i.e. reg. A
INX H	Get addr. of first number 2001H into HL pair
MOV A,M	Move the first number from memory to Accumulator i.e. reg. A
DCR C	Decrement count by 1
LOOP: INX H	Load/Get addr. of next number into HL pair
CMP M	Compare previous number with next number i.e. [A] - [[HL]]
JC Go	Jump if no carry to label Go, as the

	smaller number is in Accumulator
MOV A,M	If carry then smaller number is in memory move to Accumulator
GO: DCR C	Decrement count by 1
JNZ LOOP	Jump if count is not zero
GO: STA 2005H	Store smaller number from Accumulator to 2005H
HLT	Stop

21. Write an ALP (Assembly Language Program) to add a series of 8-bit numbers, whose sum is also 8-bit.

→ Let us assume:

Count of series at 2000H =05H
First 8-bit number at 2001H = 12H
Second 8-bit number at 2002H = 35H
Third 8-bit number at 2003H = 21H
Fourth 8-bit number at 2004H = 40H
Fifth 8-bit number at 2005H = 55H
Result at 2006H = FDH

Program	Comments
LXI H,2000H	Load/Get addr. of count 2000H into HL pair
MOV C,M	Move the count from memory to reg. C
MVI A,00H	Move initial sum 00H into Accumulator
LOOP: INX H	Load/Get addr. of next number into HL pair
ADD M	Add previous sum with next number i.e. $[A] \leftarrow [A] + [[HL]]$

GO: DCR C	Decrement count by 1
JNZ LOOP	Jump if count is not zero
GO: STA 2006H	Store result of addition from Accumulator to 2006H
HLT	Stop

22. Write an ALP (Assembly Language Program) to add two 8-bit numbers, whose sum is also 8-bit, but the result is in decimal.

→ Let us assume:

First 8-bit number at 2000H = 83H

Second 8-bit number at 2001H = 34H

result at 2002H = B7H

Converting Hexadecimal result into Decimal result:

B7H=(?)₁₀

=7*16⁰ + B*16¹

=7*1+11*16

=7+ 176

=(183)₁₀

Program	Comments
LXI H,2000H	Load/Get addr. of first number 2000H into HL pair
MOV A,M	Move the first number from memory to Accumulator i.e. reg. A
INX H	Get addr. of second number 2001H into HL pair
ADD M	Subtract second number from first number i.e. $[A] \leftarrow [A] - [[HL]]$
DAA	Decimal Adjust Accumulator i.e. convert hex result into decimal.
STA 2002H	Store result from Accumulator to 2002H
HLT	Stop

23. Write an ALP (Assembly Language Program) to add two 8-bit

numbers, whose sum is more than 8-bit, but the result is in decimal.

→ Let us assume:

1 1

First 8-bit number at 2000H

= 99H

Second 8-bit number at 2001H

= 89H

+-----

result at 2002H & 2003H

= 01H 2 2H

MSB result at 2003H =01H & LSB result at 2002H=22H

Converting Hexadecimal result into Decimal result:

122H=(?)₁₀

= $2 \times 16^0 + 2 \times 16^1 + 1 \times 16^2$

= $2 \times 1 + 2 \times 16 + 1 \times 256$

=2+ 32+ 256

=(290)₁₀

Program	Comments
MVI C,00H	Initialize reg. C for carry
LXI H,2000H	Load/Get addr. of first number 2000H into HL pair
MOV A,M	Move the first number from memory to Accumulator i.e. reg. A
INX H	Get addr. of second number 2001H into HL pair
ADD M	Subtract second number from first number i.e. [A] ← [A] - [[HL]]
DAA	Convert hex result into decimal
JNC go	Jump if no carry
INR C	If carry, increment reg. C by 1
GO:STA 2002H	Store result from Accumulator to 2002H
HLT	Stop

24. Write an ALP (Assembly Language Program) to arrange a series of 8-bit numbers in ascending order.

→ Let us assume: series of number=07H

First counter in reg. C= 06H & second counter in reg.B=06H

First 8-bit number at 2001H = 12H A=12

Second 8-bit number at 2002H = 35H

Third 8-bit number at 2003H = 21H

Fourth 8-bit number at 2004H = 40H

Fifth 8-bit number at 2005H = 55H

sixth 8-bit number at 2006H =87

seventh 8-bit number at 2007H =45

Result at:

Program	Comments
MVI C,06H	Move the first counter count in reg.C
MOV B,C	Move the second counter count in reg. B
LXI H,2001H	Load/Get addr. of first number in HL pair
LOOP: MOV A,M	Move the next no. from memory into Accumulator i.e. reg.A
INX H	Get addr. of next number in HL pair
MOV D,M	
CMP D	Compare first number with second number [A]-[D]
JC GO	Jump if carry, because smaller no. is in Accumulator
MOV A,D	If no carry, then smaller no. is in reg. D, so move the smaller number from memory to Accumulator
DCX H	Decrement HL pair by 1
MOV M,D	Store the smaller from reg, D into Memory
INX H	Increment HL pair by 1
GO:DCR C	Decrement the First counter by 1
JNZ LOOP	Jump till all comparison is done
HLT	

Program	Comments
LXI H,2000H	Load/Get addr. of count 2000H into HL pair
MOV C,M	Move the count of series from memory location 2000H to reg. C
MVI A,FFH	Move largest value FFH into Accumulator for comparison.
LOOP:INX H	Get addr. of next number into HL pair

CMP M	Compare the numbers: Subtract next number from previous number from Accumulator $[A] \leftarrow [A] - [[HL]]$
JC GO	Jump if no carry, because smaller number is in accumulator
MOV A,M	If no carry, than move the smaller number from memory to accumulator
GO:DCR C	Store result from Accumulator to 2002H
JNZ LOOP	
HLT	Stop

Practice Programs:

1] Let us assume the no. at F050H=56H

What will be the result at F051H , Reg. D and Reg.A

Program	Comments
MVI D,00H	Initialize reg. D to 00H
LXI H,F050H	Load addr. of no. in HL pair
MOV A,M	Move no. from memory to Accumulator
CMA	Take 1's complement of no.
INR A	Increment the content of reg. A by 1
MOV D,A	Move the result from Accumulator to reg. D
STA F051H	Store result from Accumulator into F051H
HLT	Stop

no. at F050H=56H

Binary of no. 56H = 0101 0110

1's complement of no. =1010 1001

result in hex = A 9 H

Add 1 = + 1

Final result at F051H, Reg. D, Reg. A = A AH

2] Let us assume no. at 5009H=33H and at 500AH=65H

find result at 500BH

Program	Comments
LXI H,5009H	Load addr. of first no. in HL pair
MOV A,M	Move first no. from memory location 5009H into reg.A
MOV B,A	Move first no. from reg. A to Reg. B
INR A	Increment contents of reg.A by 1

INX H	Load address of second no. in HL pair
ADD B	[A]+[B]=[A] i.e. 34+33=67H
STA 500BH	Store result from reg. A to 500BH
HLT	stop

A=33

B=A=33

A=A+1=33+1=34

A=34

+ B=33

result=67H

3] Let no. at 2000H and 2001H be 67H and FFH. Find result at 2002H.

Program	Comments
LXI H,2000H	Load addr. of first no. in HL pair
MOV A,M	Move first no. from memory location 2000H into reg.A
INX H	Get addr. of second no. in

	with carry.
INR A	Increment the contents of reg. A by 1
HLT	stop

A=00H=0000 0000

CF A7 A6 A5 A4 A3 A2 A1 A0

1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1

1

result CF=0 0 0 0 0 0 0 0 1 reg. A=01

INR reg. A + 1

final result= 0 0 0 0 0 0 1 0 reg. A=02H

5] Find the output of the given program: Let the no. at 2000H=FF and at 2001H=00H

Program	Comments
LXI H,2000H	Load addr. of first no. in HL pair
MOV A,M	Move first no. from memory location 2000H into reg.A
INX H	Get addr. of second no. in HL
ADD M	Add first no. with second no. [A]=[A]+[[HL]]
CMA	Take 1's complement of reg. A contents
STA 2002H	Store result from reg. A at 2002H
HLT	stop

Reg. A=FFH

+ M=00H

Result=FFH= 1111 1111

1's complement=0000 0000=00H

Final result at 2002H=00H

6] Find the result of given program at 2000H, 2001H and 2002H, if the numbers at 2000H=65H and at 2001H=4AH

Program	Comments
LXI H,2000H	Load addr. of first no. in HL pair
MOV A,M	Move first no. from memory location 2000H into reg.A
INX H	Get addr. of second no. in HL
SUB M	Subtract second no. from first no. [A]=[A]-[[HL]]
DCX H	
MOV M,A	
INR A	
STA 2002H	
HLT	

5 15

reg. A=65H

- M=4AH

Result= 15H=reg. A=2000H

+ 1

final result =16H=Reg.A=2002H

2001=4AH



