ASSEMBLY LANGUAGE PROGRAMMING

For 8085 microprocessor

EXPERIMENT 3

- **AIM:-** Write a program using register indirect addressing for loading and storing data.(data from 4150 to 4160)

- **THEORY:-**

1) Initialize HL pair as memory pointer for source.
2) Initialize DE pair as memory pointer for destination .
3) Move data from source(M) to A.
4) Store data from accumulator to destination(DL) pair.
5) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		LXI H,4150	21	Load source address in HL pair
4201			50	
4202			41	
4203		LXI D,4160	11	Load destination address in DE pair
4204			60	
4205			41	
4206		MOV A,M	7E	Copy element to Accumulator
4207		STAX D	12	Store the element to the address in the DE pair
4208				
4209		HLT	76	Program ends

RESULT:-

Input at:- 4150 : 29

Output at:- 4160 : 29

EXPERIMENT 4

- **AIM:-** WAP to add 2,8-bit numbers 28 and 49 using immediate addressing mode and store result at 4150 .

- **THEORY:-**

1) Load 28 in A using immediate addressing mode.
2) Load 49 in register B using immediate addressing mode.
3) Add value of B register with accumulator.
4) Store added value from a to 4150.
5) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		MVI A,28	21	Move 28 in accumulator
4201			28	
4202		MVI B,49	41	Move 49 in B register
4203			49	
4204		ADD B	80	Add value of B with value of A
4205		STA 4150	32	Store the value from A to the address
4206			50	
4207			41	
4208		HLT	76	Program ends

RESULT:-

Input at:- 4201 : 28

4201 : 49

Output at:- 4150 : 72

EXPERIMENT 5

- **AIM:-** WAP to add 2, 8-bit numbers in memory and store the result in memory.(without carry)

- **THEORY:-**

1) Load first operand in A.
2) Copy first operand from A to B.
3) Load second operand in A.
4) Add value of B register with accumulator.
5) Store added value from a to 4152.
5) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		LDA 4150	3A	Load data from memory to accumulator
4201			50	
			41	
4202		MOV B,A	47	Move data from A to B
4203		LDA 4151	3A	Load data from memory to accumulator
			51	
			41	
4204		ADD B	80	Add value of B with value of A
4205		STA 4152	32	Store the value from A to the address
4206			52	
4207			41	
4208		HLT	76	Program ends

RESULT:-

Input at:- 4150 : 23

4151 : AA

Output at:- 4152 : CD

EXPERIMENT 6

- **AIM:-** WAP to add 2, 8-bit numbers (using carry).

- **THEORY:-**

1) Initialize the carry as ‘Zero’
2) Load the first 8 bit data into the accumulator
3) Copy the contents of accumulator into the register ‘B’
4) Load the second 8 bit data into the accumulator.
5) Add the 2 - 8 bit data's and check for carry.
6) Jump on if no carry
7) Increment carry if there is
8) Store the added request in accumulator
9) More the carry value to accumulator
10) Store the carry value in accumulator
11) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		MVI C, 00	0E	Initialize the carry as zero
4201			00	
4202		LDA 4300	3A	Load the first 8 bit data
4203			00	
4204			43	
4205		MOV B,A	47	Copy the value into register B
4206		LDA 4301	3A	Load the second 8 bit data into the accumulator
4207			01	
4208			43	
4209		ADD B	80	Add the two values
420A		JNC	D2	Jump on if no carry
420B			OE	
420C			41	

PROGRAM:-

420D		INR C	0C	If carry , increment it by one
420E	Loop	STA 4302	32	Store the added value in the accumulator
420F			02	
4210			43	
4211		MOV A,C	79	Move the value of carry to the accumulator from register C
4212		STA 4303	32	Store the value of carry in the accumulator
4213			03	
4214			43	
4215		HLT	76	Program ends

RESULT:-

	Without Carry			With Carry
Input at:-	4300	:	23	4300 : FF
	4301	:	AA	4301 : FF
Output at:-	4302	:	CD	4302: FE
				4303 : 01(carry)

EXPERIMENT 7

- **AIM:-** WAP to subtract 2,8-bit numbers 23 from AE using immediate addressing mode and store result at 4150 .

- **THEORY:-**

1) Load AE in A using immediate addressing mode.
2) Load 23 in register B using immediate addressing mode.
3) Subtract value of B register with accumulator.
4) Store added value from a to 4150.
5) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		MVI A,AE	21	Move AE in accumulator
4201			28	
4202		MVI B,23	41	Move 49 in B register
4203			49	
4204		SUB B	90	Subtract values
4205		STA 4150	32	Store the value from A to the address
4206			50	
4207			41	
4208		HLT	76	Program ends

RESULT:-

Input at:- 4201 : AE

4201 : 23

Output at:- 4150 : 8B

EXPERIMENT 8

- **AIM:-** WAP to subtract 2, 8-bit numbers in memory and store the result in memory.(without carry)

- **THEORY:-**

1) Load first operand in A.
2) Copy first operand from A to B.
3) Load second operand in A.
4) Subtract value of B register with accumulator.
5) Store added value from A to 4152.
5) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		LDA 4150	3A	Load data from memory to accumulator
4201			50	
			41	
4202		MOV B,A	47	Move data from A to B
4203		LDA 4151	3A	Load data from memory to accumulator
			51	
			41	
4204		SUB B	90	Subtract values
4205		STA 4152	32	Store the value from A to the address
4206			52	
4207			41	
4208		HLT	76	Program ends

RESULT:-

Input at:- 4150 : AE

4151 : 23

Output at:- 4152 : 8B

EXPERIMENT 9

- **AIM:-** WAP to Subtract 2 ,8-bit numbers.(using carry)

- **THEORY:-**

1) Initialize the carry as ‘Zero’
2) Load the first 8 bit data into the accumulator
3) Copy the contents of accumulator into the register ‘B’
4) Load the second 8 bit data into the accumulator.
5) Add the 2 - 8 bit data's and check for borrow.
6) Jump on if no borrow
7) Increment borrow if there is
8) 2’s compliment of accumulator is found out
9) Store the result in the accumulator
10) Move the borrow value from ‘C’ to accumulator
11) Store the borrow value in the accumulator
12) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		MVI C 00	0E	Initialize the carry as zero
4201			00	
4202		LDA 4300	3A	Load the first 8 bit data
4203			00	
4204			43	
4205		MOV B,A	47	Copy the value into register B
4206		LDA 4301	3A	Load the second 8 bit data into the accumulator
4207			01	
4208			43	
4209		SUB B	80	Subtract the two values
420A	Loop	JNC	D2	Jump on if no borrow
420B			OE	
420C			41	

PROGRAM:-

420D		INR C	0C	If borrow is there , increment it by one
420E	Loop	CMA	2F	Compliment of 2nd data
420F		ADI , 01	06	Add one to 1`s compliment of 2nd data
4210			01	
4211		STA 4302	32	Store the result in the accumulator
4212			02	
4213			43	
4214		MOV A,C	79	Move the value of carry to the accumulator from register C
4215		STA 4303	32	Store the value of carry in the accumulator
4216			03	
4217			43	
4218		HLT	76	Program ends

RESULT:-

	Without borrow	With Carry borrow
Input at:-	4300 : 05	4300 : 07
	4301 : 07	4301 : 05
Output at:-	4302 : 02	4302: 02
		4303 : 01(borrow)

EXPERIMENT 10

- **AIM:-**Write a program to perform one's and two's complement of a number.

- **THEORY:-**

1) Load data in accumulator.
2) Find 1`s compliment .
3) Add one to find 2`s compliment.
4) Store data from accumulator to destination.
5) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		LDA 4150	3A	Load data in accumulator
4201			50	
4202			41	
4203		CMA	2F	Find 1`s compliment
4204		STA 4151		Store result
4205		ADI ,01	06	Add 1 to find 2`s compliment
4206			01	
4207		STA, 4152	32	Store result at memory address.
4208			51	
4209			41	
420A		HLT	76	Program ends

RESULT:-

Input at:- 4150 : 8F

Output at:- 4151 : 70

4152 : 71

EXPERIMENT 11

- **AIM:-**. Write a program to perform Logical AND,OR,XOR,NOT operations using logical instructions.

- **THEORY:-**

1) Load data in accumulator.
2) Move data in B register.
3) Load second data in accumulator.
4) XOR two data.
4) Store data from accumulator to destination.
5) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		LDA 4150	3A	Load first data in accumulator
4201			50	
4202			41	
4203		MOV B,A	47	Move data in B
4204		LDA 4151	3A	Load second data in accumulator.
4205			51	
4206			41	
		XRA B	A8	XOR two data
4207		STA, 4152	32	Store result at memory address.
4208			52	
4209			41	
420A		HLT	76	Program ends

RESULT:-

Input at:- 4150 : 43

4151 : 64

Output at:- 4152 : 27

EXPERIMENT 12

- **AIM:-** Write a program to perform addition of two 16-bit numbers.

- **THEORY:-**

1) Get the 1st 8 bit in ‘C’ register (LSB) and 2nd 8 bit in ‘H’ register (MSB) of 16 bit number.
2) Save the 1st 16 bit in ‘DE’ register pair .
3) Similarly get the 2nd 16 bit number and store it in ‘HL’ register pair.
4) Get the lower byte of 1st number into ‘L’ register
5) Add it with lower byte of 2nd number
6) Store the result in ‘L’ register
7) Get the higher byte of 1st number into accumulator

8) Add it with higher byte of 2nd number and carry of the lower bit addition.

9) Store the result in ‘H’ register

10) Store 16 bit addition value in ‘HL’ register pair

11) Stop program execution

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		MVI C 00	0E	Initialize the carry as zero
4201			00	
4202		LDA 4300	3A	Load the first 8 bit data
4203			00	
4204			43	
4205		MOV B,A	47	Copy the value into register B
4206		LDA 4301	3A	Load the second 8 bit data into the accumulator
4207			01	
4208			43	
4209		SUB B	80	Subtract the two values
420A	Loop	JNC	D2	Jump on if no borrow
420B			OE	
420C			41	

PROGRAM:-

420D		INR C	0C	If borrow is there , increment it by one
420E	Loop	CMA	2F	Compliment of 2nd data
420F		ADI , 01	06	Add one to 1`s compliment of 2nd data
4210			01	
4211		STA 4302	32	Store the result in the accumulator
4212			02	
4213			43	
4214		MOV A,C	79	Move the value of carry to the accumulator from register C
4215		STA 4303	32	Store the value of carry in the accumulator
4216			03	
4217			43	
4218		HLT	76	Program ends

RESULT:-

	Without carry			With Carry
Input at:-	4300	:	05	4300 : 07
	4301	:	07	4301 : 05
Output at:-	4302	:	02	4302: 02
				4303 : 01(carry)

EXPERIMENT 13

- **AIM:-** Write a program to perform subtraction of two 16-bit numbers.

- **THEORY:-**

1) Get the 1st 16 bit in 'HL' register pair
2) Save the 1st 16 bit in 'DE' register pair
3) Get the 2nd 16 bit number in 'HL' register pair
4) Get the lower byte of 1st number
5) Get the subtracted value of 2nd number of lower byte by subtracting it with lower byte of 1st number
6) Store the result in 'L' register
7) Get the higher byte of 2nd number
8) Subtract the higher byte of 1st number from 2nd number with borrow
9) Store the result in 'HL' register
10) Stop the program execution

Address	Label	Mnemonics	OP Code	Comments
4500		MVI C,00	0E	C = 00_H
4501			00	
4502		LHLD 4800	2A	L register – 1st No.
4503			00	
4504			48	
4505		XLHG	EB	Exchange contents of HL and DE register
4506		LHLD 4802	2A	HL – 2nd No.
4507			02	
4508			48	
4509		MOV A,E	7B	LSB of '1st no' to 'A'
450A		SUB L	95	A =A – L
450B		STA 4804	32	A –> memory location 4804
450C			04	
450D			48	
450E		MOV A,D	7A	MSB of 1st no to A
450F		SBB H	9C	A- A – H
4510		STA 4805	32	A – memory
4511			05	
4512			48	
4513		HLT	76	Stop execution

RESULT:-

Input: Without borrow

Input Address	Value
4800	07
4801	08
4802	05
4803	06

Output:

Output Address	Value
4804	02
4805	02
4807	00

With borrow:

Input Address	Value
4800	05
4801	06
4802	07
4803	08

Output Address	Value
4804	02
4805	02
4806	01

Calculation :

05	06	-	07	08				
05	06	0101	0110		07	08	0111	1000
CMA		1010	1001		CMA		1000	0111
ADI		0000	0001		ACI		0000	0001
		---					---	
		1010	1010				1000	1000

05	06	+	07	08	
			1010	1010	
			1000	1000	
			---		
		(1)	0010	0010	
			02	02	

EXPERIMENT 14

- **AIM:-** Write a program to perform multiplication of two 8-bit numbers.

- **THEORY:-**

1. Get the 1st 8 bit numbers
2. Move the 1st 8it number to register ‘B’
3. Get the 2nd 8 bit number
4. Move the 2nd 8 bit number to register ‘C’
5. Intialise the accumulator as zero
6. Intialise the carry as zero
7. Add both register ‘B’ value as accumulator
8. Jump on if no carry
9. Increment carry by 1 if there is carry.
10. Decrement the 2nd value and repeat from step 8, till the 2nd value become zero.

.

11. Store the multiplied value in accumulator.
12. Move the carry value to accumulator’
13. Store the carry value in accumulator

Address	Label	Mnemonics	Hex Code	Comments
4100		LDA 4500	3A	Load the first 8 bit number
4101			00	
4102			45	
4103		MOV B,A	47	Move the 1st 8 bit data to register 'B'
4104		LDA 4501	3A	Load the 2nd 16 it number
4105			01	
4106			45	
4107		MOV C,A	4F	Move the 2nd 8 bit data to register 'C'
4108		MVI A, 00	3E	Intialise the accumulator as zero
4109			00	
410A		MVI D, 00	16	Intialise the carry as zero
410B			00	
410C		ADD B	80	Add the contents of 'B' and accumulator
410D		INC	D2 11, 41	Jump if no carry
4110		INR D	14	Increment carry if there is
4111		DCR C	OD	Decrement the value 'C'

4112		JNZ 410C	C2	Jump if number zero
4113			0C	
4114			41	
4115		STA 4502	32	Store the result in accumulator
4116			02	
4117			45	
4118		MOV A,D	7A	Move the carry into accumulator
4119		STA 4503	32	Store the result in accumulator
411A			03	
411B			45	
411C		HLT	76	Stop the program execution

Result :-

Input

Input Address	Value
4500	04
4501	02

Output

Output Address	Value
4502	08
4503	00

EXPERIMENT 15

- **AIM:-** Write a program to perform division of two 8-bit numbers.

- **THEORY:-**

1. Intialise the Quotient as zero
2. Load the 1st 8 bit data
3. Copy the contents of accumulator into register ‘B’
4. Load the 2nd 8 bit data
5. Compare both the values
6. Jump if divisor is greater than dividend
7. Subtract the dividend value by divisor value
8. Increment Quotient
9. Jump to step 7, till the dividend becomes zero
10. Store the result (Quotient) value in accumulator
11. Move the remainder value to accumulator
12. Store the result in accumulator

PROGRAM:-

Address	Label	Mnemonics	Op-Code	Comments
4100		MVI C, 00	0E	Intialise Quotient as zero
4101			00	
4102		LDA, 4500	3A	Get the 1st data
4103			00	
4104			45	
4105		MOV B,A	47	Copy the 1st data into register 'B'
4106		LDA, 4501	3A	Get the 2nd data
4107			01	
4108			45	
4109		CMP B	B8	Compare the 2 values
410A		JC (LDP)	DA	Jump if dividend lesser than divisor
410B			12	
410C			41	
410D	Loop 2	SUB B	90	Subtract the 1st value by 2nd value

410E		INR C	0C	Increment Quotient (410D)
410F		JMP (LDP, 41)	C3	Jump to Loop 1 till the value of dividend becomes zero
			0D	
			41	
4112	Loop 1	STA 4502	32	Store the value in accumulator
			02	
			45	
4115		MOV A,C	79	Move the value of remainder to accumulator
4116		STA 4503	32	Store the remainder value in accumulator
4117			03	
4118			45	
4119		HLT	76	Stop the program execution

Result :-

Input

Input Address	Value
4500	09
4501	02

Output

Output Address	Value
4502	04 (quotient)
4503	01 (remainder)

Calculation :

1001
0010 – I

0111
0010 – II

0101
0010 – III

0011
0010 – IV

0001 –remainder

Quotient - 04
Remainder - 01

EXPERIMENT 16

- **AIM:-** Write a program to perform multiplication of two 16-bit numbers.

- **THEORY:-**

1. Load 16 bit data in HL pair and move data from HL pair to Stack Pointer.
2. Load another 16 bit data in HL pair and move the data to DE pair.
3. Move data 0000H to BC and HL pair.
4. Add 16 bit data present in Stack Pointer with HL pair.
5. If carry present goto Step 8 else goto step 7.
6. Increment BC register pair content once.
7. Decrement DE register pair content once.
8. Move D register content to accumulator and OR function it with E register content.

9. Check whether A is zero or not. If A=0 goto Step 6 else goto Step 5.
10. Store HL pair content in memory.
11. Move BC pair content to HL pair and then to memory.
12. Terminate the program.

Address	Label	Mnemonics	Op-code	Comments
4100		LHLD 4200	2A	Load 16 bit data from memory to HL pair
4101			00	
4102			42	
4103		SPHL	F9	Move HL pair content to stack pointer
4104		LHLD 4202	2A	Load another 16 bit data from memory to accumulator
4105			02	
4106			42	
4107		XCHG	EB	Move HL pair content to DE pair
4108		LXI H, 0000H	21	Move data 0000H to HL pair
4109			00	
410A			00	
410B		LXI B, 0000H	01	Move data 0000H to BC pair
410C			00	
410D			00	
410E	Loop1:	DAD SP	39	Add SP data with HL pair data
410F		JNC Loop2	D2	If carry present jump to specified memory
4110			13	
4111			41	
4112		INX B	03	Increment BC pair content once

4113	Loop2:	DCX D	1B	Decrement DE pair content once
4114		MOV A, D	7A	Move D register content to Acc.
4115		ORA E	B3	OR function Accumulator content with E register content
4116		JNZ Loop1	C2	Jump when no zero to specified memory
4117			0E	
4118			41	
4119		SHLD 4500	22	Store HL pair content in specified memory
411A			00	
411B			45	
411C		MOV H, B	60	Move B register content to H register
411D		MOV L, C	69	Move C register content to L register
411E		SHLD 4502	22	Store HL pair content in specified memory
411F			02	
4120			45	
4121		HLT	76	Halt

RESULT :-

INPUT DATA:		**OUTPUT DATA:**	
4200:	22	4500:	C6
4201:	22	4501:	92
4202:	33	4502:	D3
4203:	33	4503:	06

EXPERIMENT 17 (a)

- **AIM:-** Write a program to perform division of two 16-bit numbers.

- **THEORY:-**

1. Intialise ‘BC’ as ‘0000’ for Quotient
2. Load the divisor in ‘HL’ pair and save it in ‘DE’ register pair
3. Load the dividend in ‘HL’ pair
4. Move the value of ‘a’ to register ‘E’
5. Subtract the content of accumulator with ‘E’ register
6. Move the content ‘A’ to ‘C’ & ‘H’ to ‘A’
7. Subtract with borrow, the content of ‘A’ with ‘D’
8. Move the value of ‘a’ to ‘H’
9. If cy = 1, go to step 12, otherwise next step
10. Increment ‘B’ register & jump to step ‘4’
11. Add both contents of ‘DC’ and ‘HL’
12. Store the remainder in memory
13. Move the content of ‘C’ to ‘L’ & ‘B’ to ‘H’
14. Store the Quotient in memory
15. Stop the program

PROGRAM:-

Address	Label	Mnemonics	Hex Code	Comments
4500		LXI B,0000	0	Intialise Quotient as '0000'
			00	
			00	
4503		LHLD 4802	2A,02,48	Load the divisor in 'HL'
			02	
			48	
4506		XCHG	EB	Exchange 'HL' and 'DE'
4507		LHLD 4800	2A	Load the dividend
			00	
			48	
450A	Loop 2	MOV A,L	7D	Move the 'L' value to 'A'
450B		SUB E	93	(A-E) =A
450C		MOV L,A	6F	A- L (A value is move t L)
450D		MOV A,H	7C	H – A (a is stored with H)
450E		SBB D	9A	Subtract 'D' from 'A'
450F		MOV H,A	67	Then A is moved to 'H'
4510		JC loop 1	DA	If cy is present go to loop 1
4511			17	
4512			45	
4513		INX B	03	Increment BC pair by 1

4514		JMP Loop2	C3	Jump to loop 2
4515			0A	
4516			45	
4517	Loop 1	DAD ‘D’	19	‘DE’ and ‘HL’ pair all added
4518		SHLD 4806	22	HL is stored in memory
4519			06	
451A			48	
451B		MOV L,C	69	Move ‘C’ register data to ‘L’
451C		MOV H,B	60	Move ‘B’ register data to ‘H’
451D		SHLD 4804	22	Store the result in ‘HL’ pair
451E			04	
451F			48	
4520		HLT	76	Stop the program

RESULT:-

Input

Input Address	Value
4800	04
4801	00
4802	02
4803	00

Output

Output Address	Value
4804	02
4805	00
4806	FE
4807	FF

EXPERIMENT 17 (b)

- **AIM:-** Write a program to perform division of a 16-bit number with a 8-bit number .

- **THEORY:-**

PROGRAM:-

Address	Label	Mnemonics	Op-code	Comments
4000		LHLD 2200H	2A	Get the dividend
4001			00	
4002			22	
4003		LDA 2202H	3A	Get the divisor
4004			02	
4005			22	
4006		MOV C,A	4F	Move contents of A register to C register
4007		LXI D,0000H	11	Quotient = 0, Load 0000H in register pair DE
4008			00	
4009			00	
400A	BACK	MOV A,L	7D	Move contents of L register to A register
400B		SUB C	91	Subtract divisor
400C		MOV L,A	6F	Save partial result
400D		JNC SKIP	D2	if CY =1 jump
400E			11	
400F			40	

4010		DCR H	25	Subtract borrow of previous subtraction
4011	SKIP	INX D	13	Increment quotient
4012		MOV A,H	7C	
4013		CPI 00H	FE	Check if dividend < divisor
4014			00	
4015		JNZ BACK	C2	if no repeat
4016			0A	
4017			40	
4018		MOV A,L	7D	
4019		CMP C	B9	
401A		JNC BACK	D2	
401B			0A	
401C			40	
401D		SHLD 2302H	22	Store the remainder
401E			02	
401F			23	
4020		XCHG	EB	
4021		SHLD 2300H	22	

4022			00	
4023			23	
4024		HLT	76	

RESULT:-

Input :- (2200H) = 60H
(2201H) = A0H
} Dividend

(2202H) = l2H Divisor

A060H/12H = 8E8H Quotient and 10H remainder

(2300H) = E8H
(2301H) = 08H
} Quotient

(2302H= 10H
(2303H) 00H
} Remainder

EXPERIMENT 18

- **AIM:-** Write a program to move a block of four 8-bit numbers from 2001 (2001-2004) to 3000 (3000-3003).

- **THEORY:-**

1) Load the DE pair with the destination address.
2) Load the HL pair with the count of elements in the data block.
3) Load element in the data block.
4) Increment the source address.
5) Copy the element to the accumulator and then transfer it to the destination address.
6) Increment destination address.
7) Decrement the count.
8) If Count = 0 then go to the next step else go to step 3.
9) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		LXI D,3000	11	Load destination address in DE pair
4201			00	
4202			30	
4203		LXI H,2000	21	Load the count in HL pair
4204			00	
4205			20	
4206		MOV C,M	4E	Copy the count to register C
4207	LOOP	INX H	23	Increment memory
4208		MOV A,M	7E	Copy element to Accumulator
4209		STAX D	12	Store the element to the address in the DE pair
420A		INX D	13	Increment destination address
420B		DCR C	OD	Decrement count
420C		JNZ LOOP	C2	Jump on non-zero to address 4207
420D			07	
420E			42	
420F		HLT	76	Program ends

RESULT:-

Input at:-	2000	:	04 (stores count of numbers)
	2001	:	06
	2002	:	07
	2003	:	12
	2004	:	03
Output at:-	3000	:	06
	3001	:	07
	3002	:	12
	3003	:	03

EXPERIMENT 19

- **AIM:-** Write a program to move a block of four 8-bit numbers from 2001 (2001-2004) to 3003 (3003-3000) i.e. in reverse order.

- **THEORY:-**

1) Load the DE pair with the destination address.
2) Load the HL pair with the count of elements in the data block.
3) Load element in the data block.
4) Increment the source address.
5) Copy the element to the accumulator and then transfer it to the destination address.
6) Decrement destination address (as elements are to be stored in reverse order).
7) Decrement the count.
8) If Count = 0 then go to the next step else go to step 3.
9) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4200		LXI D,3003	11	Load destination address in DE pair
4201			03	
4202			30	
4203		LXI H,2000	21	Load the count in HL pair
4204			00	
4205			20	
4206		MOV C,M	4E	Copy the count to register C
4207	LOOP	INX H	23	Increment memory
4208		MOV A,M	7E	Copy element to Accumulator
4209		STAX D	12	Store the element to the address in the DE pair
420A		DCX D	1B	Increment destination address
420B		DCR C	OD	Decrement count
420C		JNZ LOOP	C2	Jump on non-zero to address 4207
420D			07	
420E			42	
420F		HLT	76	Program ends

RESULT:-

Input at:- 2000 : 04 (stores count of numbers)

2001 : 06

2002 : 07

2003 : 12

2004 : 03

Output at:- 3003 : 06

3002 : 07

3001 : 12

3000 : 03

EXPERIMENT 20

- **AIM:-** Write a program to sort given 'n' numbers in ascending order.

- **THEORY:-**

1) Initialize HL pair as memory pointer.
2) Get the count at 4200 in to C register.
3) Copy it in D register.
4) Get the first vale in Accumulator.
5) Compare it with the value at next location.
6) If they are out of order, exchange the contents of accumulator and memory.
7) Decrement D register's content by 1.
8) Repeat steps 5 and 7 till the value in D register become zero.
9) Decrement C register's content by 1.
10) Repeat steps 3 to 9 till the value in C register becomes zero.
11) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4400		LXI H,4200	21	Load the array size to the HL pair
4401			00	
4402			42	
4403		MOV C,M	4E	Copy the array size to C register
4404		DCR C	0D	Decrement C by 1
4405	REPEAT	MOV D,C	51	Copy content of C to D register
4406		LXI H,4201	21	Load the first data to the HL pair
4407			01	
4408			42	
4409	LOOP	MOV A,M	7E	Copy the data to the accumulator
440A		INX H	23	Increment memory by 1
440B		CMP M	BE	Compare accumulator and memory content
440C		JC SKIP	DA	Jump on carry to the address 4414
440D			14	
440E			44	
440F		MOV B,M	46	Copy memory content to B register
4410		MOV M,A	77	Copy accumulator content to memory
4411		DCX H	2B	Decrement memory by 1
4412		MOV M,B	70	Copy B register’s content to memory
4413		INX H	23	Increment memory by 1
4414	SKIP	DCR D	15	Decrement D by 1

4415		JNZ LOOP	C2	Jump on non-zero to the address 4409
4416			09	
4417			44	
4418		DCR C	0D	Decrement C by 1
4419		JNZ REPEAT	C2	Jump on non-zero to the address 4405
441A			05	
441B			44	
441C		HLT	76	Program ends

Result:-

```
Input at:-      4200   :       05 ---------- Array Size
                4201   :       05
                4202   :       04
                4203   :       03
                4204   :       02
                4205   :       01

Output at:-     4200   :       05 ---------- Array Size
                4201   :       01
                4202   :       02
                4203   :       03
                4204   :       04
                4205   :       05
```

EXPERIMENT 21

- **AIM:-** Write a program to sort given 'n' numbers in descending order.

- **THEORY:-**

1) Initialize HL pair as memory pointer.
2) Get the count at 4200 in to C register.
3) Copy it in D register.
4) Get the first vale in Accumulator.
5) Compare it with the value at next location.
6) If they are out of order, exchange the contents of accumulator and memory.
7) Decrement D register's content by 1.
8) Repeat steps 5 and 7 till the value in D register become zero.
9) Decrement C register's content by 1.
10) Repeat steps 3 to 9 till the value in C register becomes zero.
11) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENT
4400		LXI H,4200	21	Load the array size to the HL pair
4401			00	
4402			42	
4403		MOV C,M	4E	Copy the array size to C register
4404		DCR C	0D	Decrement C by 1
4405	REPEAT	MOV D,C	51	Copy content of C to D register
4406		LXI H,4201	21	Load the first data to the HL pair
4407			01	
4408			42	
4409	LOOP	MOV A,M	7E	Copy the data to the accumulator
440A		INX H	23	Increment memory by 1
440B		CMP M	BE	Compare accumulator and memory content
440C		JNC SKIP	DA	Jump on no carry to the label SKIP
440D			14	
440E			44	
440F		MOV B,M	46	Copy memory content to B register
4410		MOV M,A	77	Copy accumulator content to memory
4411		DCX H	2B	Decrement memory by 1
4412		MOV M,B	70	Copy B register’s content to memory
4413		INX H	23	Increment memory by 1
4414	SKIP	DCR D	15	Decrement D by 1

4415		JNZ LOOP	C2	Jump on non-zero to the label LOOP
4416			09	
4417			44	
4418		DCR C	0D	Decrement C by 1
4419		JNZ REPEAT	C2	Jump on non-zero to the label REPEAT
441A			05	
441B			44	
441C		HLT	76	Program ends

Result:-

Input at:- 4200 : 05 ---------- Array Size
4201 : 01
4202 : 02
4203 : 03
4204 : 04
4205 : 05

Output at:- 4200 : 05 ---------- Array Size
4201 : 05
4202 : 04
4203 : 03
4204 : 02
4205 : 01

EXPERIMENT 22

- **AIM:-** Write a program to calculate sum of 'n' elements (n numbers are stored at consecutive memory locations in an array calculate their sum assume sum comes in 16 bit take numbers in array such that sum is 16 bit).

- **THEORY:-**

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENTS
4100		LXI H,4150	21	Point to table base
4101			50	
4102			41	
4103		MOV B,M	46	Get length of counter
4104		INX H	23	Point to first entry
4105		MVI E,00H	1E	Initialize E register
4106			00	
4107		XRA A	AF	Clear accumulator
4108	LOOP 1	ADD M	86	Add contents of memory with data in accumulator
4109		JNC LOOP 2	D2	If no carry jump to previous address
410A			0D	
410B			41	
410C		INR E	1C	Increment register E
410D	LOOP 2	INX H	23	Increment H-L pair to point to next location in array
410E		DCR B	05	Decrement counter
410F		JNZ LOOP 1	C2	If counter not zero jump to address 4108
4110			08	
4111			41	
4112		STA 4140	32	Store contents of accumulator i.e. sum stored in accumulator at memory location 4140
4113			40	
4114			41	

4115		MOV A,E	7B	
4116		STA 4141	32	
4117			41	
4118			41	
4119		HLT	76	Halt

Result:-

Input at:- 4150: 06 (Counter)
4151: 54
4152: 63
4153: 7A
4154: 2B
4155: 8F
4156:FF

Output at:- 4140: EA
4141:02

54+63+7A+2B+8F+FF=02EA

EXPERIMENT 23

- **AIM:-** Write a program to perform addition of two arrays of data each containing 10 numbers and store result in third block.

[2200] + [2300]=[2400]

! ! !

[221A] + [231A]=[241A]

- **THEORY:-**

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENTS
4400		LXI H, 2200	21	Initialize memory pointer 1
4401			00	
4402			22	
4403		LXI B,2300	4E	Initialize memory pointer 2
4404			00	
4405			23	
4406		LXI D,2400	23	Initialize result pointer
4407			00	
4408			24	
4409	BACK	LDAX B	0A	Get the number from array 2
440A		ADD M	86	Add it with number in array 1
440B		STAX D	12	Store the addition in array 3
440C		INX H	23	Increment pointer 1
440D		INX B	03	Increment pointer 2
440E		INX D	13	Increment result pointer
440F		MOV A,L	7D	
4410		CPI 0A	FE	Check pointer 1 for last number
4411			0A	
4412		JNZ BACK	C2	If not zero jump to location 4409
			09	
			44	
4412		HLT	76	Halt

Result:-

Input at:-

Output at:-

EXPERIMENT 24

- **AIM:-** Write a program to perform factorial of an 8-bit number.

- **THEORY:-**

1) Load HL pair with the number whose factorial is to be found.
2) Copy memory content to register E.
3) Decrement register B content by 1.
4) Increment the memory and decrement the number and store the numbers in the consecutive memory locations.
5) Again load the number to the HL pair.
6) Copy the number to the accumulator.
7) Increment memory and copy the second number to register B.
8) Copy the accumulator content to the register C as count.
9) Initialize accumulator with 0.
10) Add accumulator content with register B.

11) Decrement count.
12) If count = 0, then got next step else go to step 8.
13) Copy accumulator content to memory
14) Decrement register E by 1.
15) If register E is not zero then go to step 6 else go to next step 8.
16) Store the factorial in memory.
17) Terminate the program.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP- CODE	COMMENTS
4200		LXI H,4100	21	Load the number to the HL pair
4201			00	
4202			41	
4203		MVI C,M	4E	Copy the number to register C
4204		MVI E,M	5E	Copy memory content to E
4205		DCR E	1D	Decrement E
4206	LOOP1	INX H	23	Increment memory
4207		DCR C	0D	Decrement C
4208		MOV M,C	71	Copy content of C to memory
4209		JNZ LOOP1	C2	Jump on non-zero to label LOOP1
420A			06	
420B			42	
420C		LXI H,4100	21	Load the number whose factorial is to be found in HL pair
420D			00	
420E			41	
420F	LOOP2	MOV A,M	7E	Copy memory content to accumulator
4210		INX H	23	Increment memory
4211		MOV B,M	46	Copy memory content to B
4212		MOV C,A	4F	Copy accumulator content to C
4213		MVI A,00	3E	Initialize A with 0
4214			00	
4215	GO	ADD B	80	[A] ← [A] + [B]

4216		DCR C	0D	Decrement C
4217		JNZ GO	C2	Jump on non-zero to label GO
4218			15	
4219			42	
421A		MOV M,A	77	Copy accumulator content to memory
421B		DCR E	1D	Decrement E
421C		JNZ LOOP2	C2	Jump on non-zero to label LOOP2
421D			0F	
421E			42	
421F		STA 4500	32	Store accumulator content to memory
4220			00	
4221			45	
4222		HLT	76	Halt

Result:-

Input at:- 4100 : 05_H

Output at:- 4500 : 78_H ---------- factorial

EXPERIMENT 25

- **AIM:-** Write a program to display Fibonacci series.

- **THEORY:-**

1. This program generates the Fibonacci series. The Fibonacci series is: 0 1 1 2 3 5 8 13 21 34.
2. In hexadecimal, it will be: 00 01 01 02 03 05 08 0D 15 22 ? The first two numbers of the series are 0 and 1.
3. The third number is computed as 0 + 1 = 1, fourth number is 1 + 1 = 2, fifth number is 1 + 2 = 3 and so on.
4. The count is initialized in register D to display the numbers in series.
5. Initialize register B to first number 00H and register C to second number 01H.
6. Initialize H-L pair to point to memory location 3000H.
7. Move the first two numbers from registers B and C to memory locations 3000H and 3001H.

8. Add the two numbers and store the result as first number.
9. Increment H-L pair and move the result from accumulator to memory location.
10. The next term is then computed by making the result equal to previous number.
11. The process is repeated until all the numbers are calculated.

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OP-CODE	COMMENTS
2000		MVI D,08H	16	Initialize counter to display numbers in series.
2001			08	
2002		MVI B,00H	06	Initialize reg. B to store previous number
2003			00	
2004		MVI C,01H	0E	Initialize reg. C to store current number.
2005			01	
2006		LXI H,3000H	21	Initialize H-L pair to point to memory.
2007			00	
2008			30	
2009		MOV M,B	70	Move 00H from reg. B to memory.
200A		INX H	23	Increment H-L pair.
200B		MOV M, C	71	Move 01H from reg. C to memory.
200C		MOV A, B	78	Move previous number from reg. B to reg. A.
200D	LOOP	ADD C	81	Add the two numbers
200E		MOV B, C	41	Assign current number to previous number.
200F		MOV C, A	4F	Save result as new current number.
2010		INX H	23	Increment H-L pair.
2011		MOV M, A	77	Move number from reg. A to memory.
2012		DCR D	15	Decrement counter.
2013		JNZ LOOP	C2	Jump to address 200DH if counter is not zero.
2014			0D	
2015			20	
2016		HLT	76	Halt

Result:-

Output:
After Execution:

3000:	00
3001:	01
3002 :	01
3003 :	02
3004 :	03
3005:	05
3006:	08
3007:	0D
3008:	15
3009:	22

EXPERIMENT 26

- **AIM:-** Write a program to perform:-

a) Binary to BCD conversion

b) BCD to Binary conversion

c) ASCII to Decimal conversion

d) Decimal to ASCII conversion

- **THEORY:-** a) Binary to BCD conversion

1. Clear ‘D’ and ‘E’ register to account for hundred’s and ten’s
2. load the binary data in accumulator
3. Compare ‘A’ with 64 if cy = 01, go step C otherwise next step
4. Subtract 64 from (64+1) ‘A’ register
5. Increment ‘E’ register
6. Compare the register ‘A’ with ‘0A’, if cy=1, go to step 11,otherwise next step.
7. Subtract ($0A_H$) from ‘A’ register

8. Increment D register
9. Go to step 7
10. Combine the units and tens to from 8 bit result
11. Save the units, tens and hundred's in memory
12. Stop the program execution

PROGRAM:-

Memory Location	Hex Code	Label	Mnemonics	Comments
4100	0E		MVI E,00	Clear 'E' register (Hund)
4101	00			
4102	53		MOV D,E	Clear 'D' register (tens)
4103	3A		LDA 4200	Get the data in 'A'
4104	00			
4105	42			
4106	C3	HUND	CPI 64	Compare the data with 64
4107	64			
4108	DA		JC TEN	If content is less jump to ten
4109	11			
410A	41			
410B	D6		SUI 64	Subtract data by 64
410C	64			
410D	IC		INR E	Increment carry each time
410E	C3		JMP HUND	Jump to hundred & repeat
410F	06			

4111	C3	TEN	CPI 0A	Compare the data with 0A
4112	0A			
4113	DA		JC UNIT	If data is less jump to unit
4114	1C			
4115	41			
4116	D6		SUI 0A	Subtract the data by 0A
4117	0A			
4118	14		INR D	Increment 'D' each time
4119	C3		JMP TEN	Jump to ten & repeat
411A	11			
411B	41			
411C	4F	UNIT	MOV C,A	Move the value 'A' to 'C'
411D	7A		MOV A,D	Move the value 'D' to 'A'
411E	07		RLC	Rotate the value of 'A'
411F	07		RLC	Of 'A' so that
4120	07		RLC	Lower and upper niddle
4121	07		RLC	Gets exchanged

4122	81		ADD C	Add ‘A’ and ‘C’
4123	32		STA 4250	Save ten’ & units in ‘M’
4124	50			
4125	42			
4126	7B		MOV A,E	Move to E to A
4127	32		STA 4251	Save hundreds unit in ‘A’
4128	51			
4129	42			
412A	76		HLT	Stop the program execution

Result:

Input

Input Address	Value
4200	54

Output

Output Address	Value
4250	84
4251	00

b) BCD to Binary conversion

1. Get the BCD data in accumulator and save it in register ‘E’
2. Mark the lower nibble of BCD data in accumulator
3. Rotate upper nibble to lower nibble and save it in register ‘B’
4. Clear the accumulator
5. Move $0A_H$ to ‘C’ register
6. Add ‘A’ and ‘B’ register
7. Decrement ‘C’ register. If zero flag = 0, go to step 7
8. Save the product in ‘B’
9. Get the BCD data in accumulator from ‘E’ register and mark the upper nibble
10. Add the units (A-ug) to product (B-ug)
11. Store the binary value in memory
12. End the program

Memory Location	Op-Code	Label	Mnemonics	Comments
4100	3A		LDA 4200	Get the data in 'A'
4101	00			
4102	42			
4103	5E		MOV E,A	Save in 'E' register
4104	E6		ANI F0	Mark the lower nibble
4105	F0			
4106	07		RLC	Rotate the upper
4107	07		RLC	To lower nibble
4108	07		RLC	And save in
4109	07		RLC	Register B
410A	47		MOV B,A	Move it from 'A' to 'B'
410B	AF		XRA A	Clear the accumulator
410C	0E		MVI C,0A	Intialise 'C' as '0A'
410D	0A			
410E	08		REP	
410F	0D		DCR C	Decrement 'C' register
4110	C2		JNZ	Jump till value 'C' is 0
4111	0E			
4112	41			

4113	47		MOV B,A	Move the value A to B
4114	7B		MOV A,E	Get the BCD in ‘A’
4115	E6		ANI 0F	Mark the upper nibble
4116	0F			
4117	80		ADD B	Add ‘A’ and ‘B’
4118	32		STA 4201	Save the binary data
4119	01			
411A	42			
411B	76		HLT	Stop the program

Input

Input Address	Value
4200	68

Output

Output Address	Value
4201	44

$$\begin{array}{r|l} 16 & 68 \\ \hline & 4\text{-}4 \end{array}$$

c) ASCII to Decimal conversion

Address	Label	Mnemonics	Op-code	Comments
4100		LDA 4200	3A	Load number to be converted in accumulator
4101			00	
4102			42	
4103		SUI 30	D6	Subtract immediate 30 from content in accumulator
4104			30	
4105		JC LABEL2	DA	If carry jump to location 410D
4106			0D	
4107			41	
4108		CPI 0A	FE	Compare immediate 0A with content in accumulator
4109			0A	
410A		JC LABEL1	DA	If carry jump to location 410F
410B			0F	
410C			41	
410D	LABEL2	MVI A,EE	3E	Move immediate data EE to accumulator
410E			EE	

410F	LABEL1	STA 4201	32	Store data at accumulator in location 4201
4110			01	
4111			42	
4112		HLT	76	Stop program execution

RESULT :-

ADDRESS	DATA	INPUT/OUTPUT
4200	36	Input
4201	06	Output

d) Decimal to ASCII conversion

Address	Label	Mnemonics	Op-code	Comments
4100		LDA 4200	3A	Load number to be converted in accumulator
4101			00	
4102			42	
4103		CPI 0A	FE	Compare immediate data 0A with accumulator
4104			0A	
4105		JC LABEL1	DA	If carry jump to location 410D
4106			0D	
4107			41	
4108		ADI 30	C6	Add immediate 30 to accumulator
4109			30	
410A		JMP LABEL2	C3	Jump to location 410F
410B			0F	
410C			41	
410D	LABEL1	MVI A,EE	3E	Move immediate data EE to accumulator
410E			EE	
410F	LABEL2	STA 4201	32	Store data in accumulator to location 4201
4110			01	
4111			42	
4112		HLT	76	

RESULT :-

ADDRESS	DATA	INPUT/OUTPUT
4200	01	Input
4201	31	Output

EXPERIMENT 27

- **AIM:-** Write a program to perform:-
  a) BCD to Hexadecimal conversion
  b) Hexadecimal to decimal conversion
  c) Hexadecimal to Binary conversion
  d) Hexadecimal to ASCII conversion

- **THEORY:-** a) BCD to Hexadecimal conversion

1. Initialize memory pointer to 4150 H
2. Get the Most Significant Digit (MSD)
3. Multiply the MSD by ten using repeated addition
4. Add the Least Significant Digit (LSD) to the result obtained in previous step
5. Store the HEX data in Memory

PROGRAM:-

Address	Label	Mnemonics	Op-code	Comments
4100		LXI H,5000		
4101			00	
4102			50	
4103		MOV A.M	7E	Initialize memory pointer
4104		ADD A	87	MSD X 2
4105		MOV B,A	47	Store MSD X 2
4106		ADD A	87	MSD X 4
4107		ADD A	87	MSD X 8
4108		ADD B	80	MSD X 10
4109		INX H	23	Point to LSD
410A		ADD M	86	Add to form HEX
410B		INX H	23	
410C		MOV M,A	77	Store the result
410D		HLT	76	

RESULT :-

Input:
Data 0: 02H in memory location 5000
Data 1: 09H in memory location 5001

Output:
Data 0: 1DH in memory location 5002

b) Hexadecimal to decimal conversion

1. Start the program.
2. Load data from memory to accumulator and move the data 00 to D and E registers.
3. Compare the accumulator data with the data 64.
4. If carry=0 jump to Step 6 else jump to Step 5.
5. Jump to Step 10.
6. Subtract accumulator data by 64.
7. Increment the content of D register once.
8. If carry=0 jump to Step 6 else jump to Step 9.
9. Decrement the content of D register once and add data 64 with accumulator.
10. Subtract accumulator data by 0A and Increment E register content once.
11. If carry=0 jump to Step 10 and Decrement E register content once.
12. Add data 64 with accumulator and move it to C register.
13. Move E register content to accumulator.
14. Rotate the accumulator content 4 times by left.
15. Add C register content with accumulator content.
16. Store data in accumulator pair to specified memory
17. Move D register content to accumulator
18. Store data in accumulator pair to specified memory.
19. End.

Memory	Label	Mnemonics		Op-code	Comments
4100		MVI	E,00H	1E	Move data 00 to E register
4101				00	
4102		MVI	D, 00H	16	Move data 00 to D register
4103				00	
4014		LDA	4200	3A	Load data from memory to
4105				00	accumulator
4106				42	
4107		CPI	64	FE	Compare the accumulator data with
4108				64	the data 64
4109		JNC	410F	D2	If carry=0 jump to specified memory
410A				0F	
410B				41	
410C		JMP	4118	C3	Jump to specified memory
410D				18	
410E				41	
410F	Loop1	SUI	64	D6	Subtract accumulator data by 64
4110				64	
4111		INR	D	14	Increment D register content once
4112		JNC	410F	D2	If carry=0 jump to specified memory
4113				0F	
4114				41	
4115		DCR	D	15	Decrement D register content once
4116		ADI	64	C6	Add data 64 with accumulator
4117				64	

4118	Loop2	SUI	0A	D6	Subtract accumulator data by 0A	
4119				0A		
411A		INR	E	1C	Increment E register content once	
411B		JNC	4118	D2	If carry=0 jump to specified memory	
411C				18		
411D				41		
411E		DCR	E	1D	Decrement E register content once	
`411F		ADI	0A	C6	Add data 64 with accumulator	
4120				0A	Move accumulator content to C	
4121		MOV	C, A	4F	register	
4122		MOV	A, E	7B	accumulator	
4123		RLC		07	Rotate the accumulator content 4	
4124		RLC		07	tines	
4125		RLC		07	by left	
4126		RLC		07	Add C register content with	
4127		ADD	C	81	Accumulator content	
4128		STA	4500	32	Store data in accumulator pair to	
4129				00	specified memory	
412A				45	Move D register content to	
412B		MOV	A, D	7A	accumulator	
412C		STA	4501	32	Store data in accumulator pair to	
412D				01	specified memory	
412E				45		
412F		HLT		76	Halt	

RESULT :-

INPUT DATA:	**OUTPUT DATA:**	
4200: CE	4500:	06
	4501:	02

c) Hexadecimal to Binary conversion

1. Initialize the memory address
2. Load the value 08 to B register
3. Load the hexadecimal number to accumulator
4. Rotate the content towards right once
5. Check for carry, if carry = 1 got to step 8 otherwise execute the next step
6. Clear the memory location
7. Jump to step 9
8. Move the 01 to memory
9. Increment the memory location
10. Decrement the B register content
11. Jump on non zero to 4 other wise stop.

ADDRESS	LABEL	MNEMONICS	Op-code	COMMENTS
4100		LXI H, 4500	21	Initialize the memory location
4101			00	
4102			45	
4103		MVI B, 08	06	Move the content 08 to B register
4104			08	
4105		MVI A, 5A	3E	Move the content 5A to A register
4106			5A	
4107	L3	RRC	0F	Rotate the content through right
4108		JC L1	DA	Jump on carry to location L1
4109			10	
410A			41	
410B		MVI M, 00	36	Move the memory location 00
410C			00	

410D		JMP L2	C3	Jump to Location L2
410C			12	
410E			41	
4110	L1	MVI M, 01	36	Move the content to 01 to Memory location
4111			01	
4112	L2	INX H	23	Increment the HL
4113		DCR B	05	Decrement to B register
4114		JNZ L3	C2	Jump on no zero to L3
4115			04	
4116			41	
4117		HLT	76	Stop

Before Execution :-

ADDRESS	DATA
4500	FF

After Execution :-

ADDRESS	DATA
4501	01
4502	01
4503	01
4504	01
4505	01
4506	01
4507	01
4508	01

d) Hexadecimal to ASCII conversion

1) Load the HEX value in to the accumulator.
2) Compare the number with 0A.
3) If number < 0A then jump to step 7.
4) Add 07_H with the number and store the result in the accumulator.
5) Store the result in memory.
6) Terminate the program.

MEMORY	LABEL	MNEMONIC	HEX CODE	COMMENT
4300		LDA 4400	3A	Load the HEX value to the accumulator
4301			00	
4302			44	
4303		CPI 0A	FE	Compare accumulator with 0A
4304			0A	
4305		JC AHEAD	DA	Jump on carry to the label AHEAD
4306			0A	
4307			43	
4308		ADI O7	C6	Add 07_H with the accumulator content
4309			07	
430A	AHEAD	ADI 30	C6	Add 30_H with the accumulator content
430B			30	
430C		STA 4401	32	Store the ASCII value in 4401
430D			01	
430E			44	
430F		HLT	76	Program ends

RESULT :-

Input at	4400	:	06	------- HEX value
Output at	4401	:	36	------- ASCII value

EXPERIMENT 28

- **AIM:-** Write a program to find largest and smallest number in an array.

- **THEORY:-** a) To find the largest element in an array.

1. Initially, the counter is initialized with the size of an array.
2. Then, two numbers are moved to registers A and B, and compared.
3. After comparison, the largest of two must be in accumulator. If it is
4. already in accumulator, then its fine, otherwise it is moved to accumulator.
5. Counter is decremented and checked whether it has reached zero. If it has, the loop terminates otherwise, the next number is moved to register and compared.
6. Let us assume that the memory location 3000H stores the counter. The next memory locations store the array.
7. Initially, H-L pair is loaded with the address of the counter and is moved to register C.
8. Then, H-L pair is incremented to point to the first number in the array.
9. The first number is moved from memory to accumulator and counter is decremented by one.
10. H-L pair is again incremented and second number is moved to register B.
11. The two numbers are compared.

12. After comparison, if A > B, then CF = 0, and if A < B, then CF = 1.
13. Carry flag is checked for carry. If there is a carry, it means B is greater than A and it is moved to accumulator.
14. Counter is decremented and checked whether it has become zero.
15. If it hasn't become zero, it means there are numbers left in the array. In this case, the control jumps back to increment the H-L pair and moves the next number to register B.
16. This process continues until counter becomes zero, i.e. all the numbers in the array are compared.
17. At last, H-L pair is incremented and the largest number is moved from accumulator to memory.

Address	Mnemonics	Operand	Opcode	Remarks
2000	LXI	H, 3000H	21	Load H-L pair with address 3000H.
2001			00	Lower-order of 3000H.
2002			30	Higher-order of 3000H.
2003	MOV	C, M	4E	Move counter from memory to reg. C.
2004	INX	H	23	Increment H-L pair.
2005	MOV	A, M	7E	Move the 1st number from memory to reg. A.
2006	DCR	C	0D	Decrement counter.
2007	INX	H	23	Increment H-L pair.
2008	MOV	B, M	46	Move the next number from memory to reg. B.
2009	CMP	B	B8	Compare B with A.
200A	JNC	200EH	D2	Jump to address 200EH if there is no carry.
200B			0E	Lower-order of 200EH.
200C			20	Higher-order of 200EH.
200D	MOV	A, B	78	Move largest from reg. B to reg. A.
200E	DCR	C	0D	Decrement counter.

200F	JNZ	2007H	C2	Jump to address 2007H if counter is not zero.
2010			07	Lower-order of 2007H.
2011			20	Higher-order of 2007H.
2012	INX	H	23	Increment H-L pair.
2013	MOV	M, A	77	Move the result from reg. A to memory.
2014	HLT		76	Halt.

Result:

Before Execution:

3000H:	05H (Counter)
3001H:	15H
3002H:	01H
3003H:	65H
3004H:	E2H
3005H:	83H

After Execution:

3006H:	E2H

b) To find the smallest element in an array.

1. Initially, the counter is initialized with the size of an array.
2. Then, two numbers are moved to registers A and B, and compared.
3. After comparison, the smallest of two must be in accumulator. If it is already in accumulator, then its fine, otherwise it is moved to accumulator.
4. Counter is decremented and checked whether it has reached zero. If it has, the loop terminates otherwise, the next number is moved to register and compared.
5. Let us assume that the memory location 3000H stores the counter. The next memory locations store the array.
6. Initially, H-L pair is loaded with the address of the counter and is moved to register C.
7. Then, H-L pair is incremented to point to the first number in the array.
8. The first number is moved from memory to accumulator and counter is decremented by one.
9. H-L pair is again incremented and second number is moved to register B.
10. The two numbers are compared.
11. After comparison, if A > B, then CF = 0, and if A < B, then CF = 1.

12. Carry flag is checked for carry. If there is no carry, it means B is smaller than A and it is moved to accumulator.
13. Counter is decremented and checked whether it has become zero.
14. If it hasn't become zero, it means there are numbers left in the array. In this case, the control jumps back to increment the H-L pair and moves the next number to register B.
15. This process continues until counter becomes zero, i.e. all the numbers in the array are compared.
16. At last, H-L pair is incremented and the smallest number is moved from accumulator to memory.

Program:

Address	Mnemonics	Operand	Opcode	Remarks
2000	LXI	H, 3000H	21	Load H-L pair with address 3000H.
2001			00	Lower-order of 3000H.
2002			30	Higher-order of 3000H.
2003	MOV	C, M	4E	Move counter from memory to reg. C.
2004	INX	H	23	Increment H-L pair.
2005	MOV	A, M	7E	Move the 1st number from memory to reg. A.
2006	DCR	C	0D	Decrement counter.
2007	INX	H	23	Increment H-L pair.
2008	MOV	B, M	46	Move the next number from memory to reg. B.
2009	CMP	B	B8	Compare B with A.
200A	JC	200EH	DA	Jump to address 200EH if there is no carry.

200B			0E	Lower-order of 200EH.
200C			20	Higher-order of 200EH.
200D	MOV	A, B	78	Move smallest from reg. B to reg. A.
200E	DCR	C	0D	Decrement counter.
200F	JNZ	2007H	C2	Jump to address 2007H if counter is not zero.
2010			07	Lower-order of 2007H.
2011			20	Higher-order of 2007H.
2012	INX	H	23	Increment H-L pair.
2013	MOV	M, A	77	Move the result from reg. A to memory.
2014	HLT		76	Halt.

Result:

Before Execution:

3000H:	05H (Counter)
3001H:	15H
3002H:	01H
3003H:	65H
3004H:	E2H
3005H:	83H

After Execution:

3006H:	01H

EXPERIMENT 29

- **AIM:-** Write a program to calculate square of a number.

- **THEORY:-**

1) Load the HL pair with the number whose square is to be found.
2) Copy the number to C and B registers.
3) Initialize D and A registers with 0.
4) Add content of accumulator with register B.
5) Check for carry.
6) If no carry go to step 8 else go to next step.
7) Increment content of D.
8) Decrement content of C.
9) If ZF = 0 then go to next step else go to step 4.
10) Store the product and carry in memory.
11) Terminate the program

PROGRAM:-

MEMORY	LABEL	MNEMONIC	OPCODE	COMMENT
4200		LXI H,4500	21	Load the number to the HL pair
4201			00	
4202			45	
4203		MVI A,M	7E	Copy the number to the accumulator
4204		MVI B,A	47	Copy the accumulator content to register B
4205		MVI D,00	16	Initialize D with 0
4206			00	
4207		MOV C,A	4F	Copy accumulator content to C
4208		MOV A,D	7A	Copy content of D to accumulator
4209	GO	ADD B	80	[A] ß [A] + [B]
420A		JNC AHEAD	D2	Jump on no carry to the label AHEAD
420B			0E	
420C			42	
420D		INR D	14	Increment D
420E	AHEAD	DCR C	0D	Decrement C
420F		JNZ GO	C2	Jump on non-zero to the label GO
4210			09	
4211			42	
4212		STA 4100	32	Store the product to 4100
4213			00	
4214			42	

4215		MOV A,D	74	Copy carry to accumulator
4216		STA 4101	32	Store carry to 4101
4217			01	
4218			41	
4219		HLT	76	Program ends

Result:-

Input at 4100 : 05_H

Output at 4100 : 19_H ---------- Square

4101 : 00_H ---------- Carry