

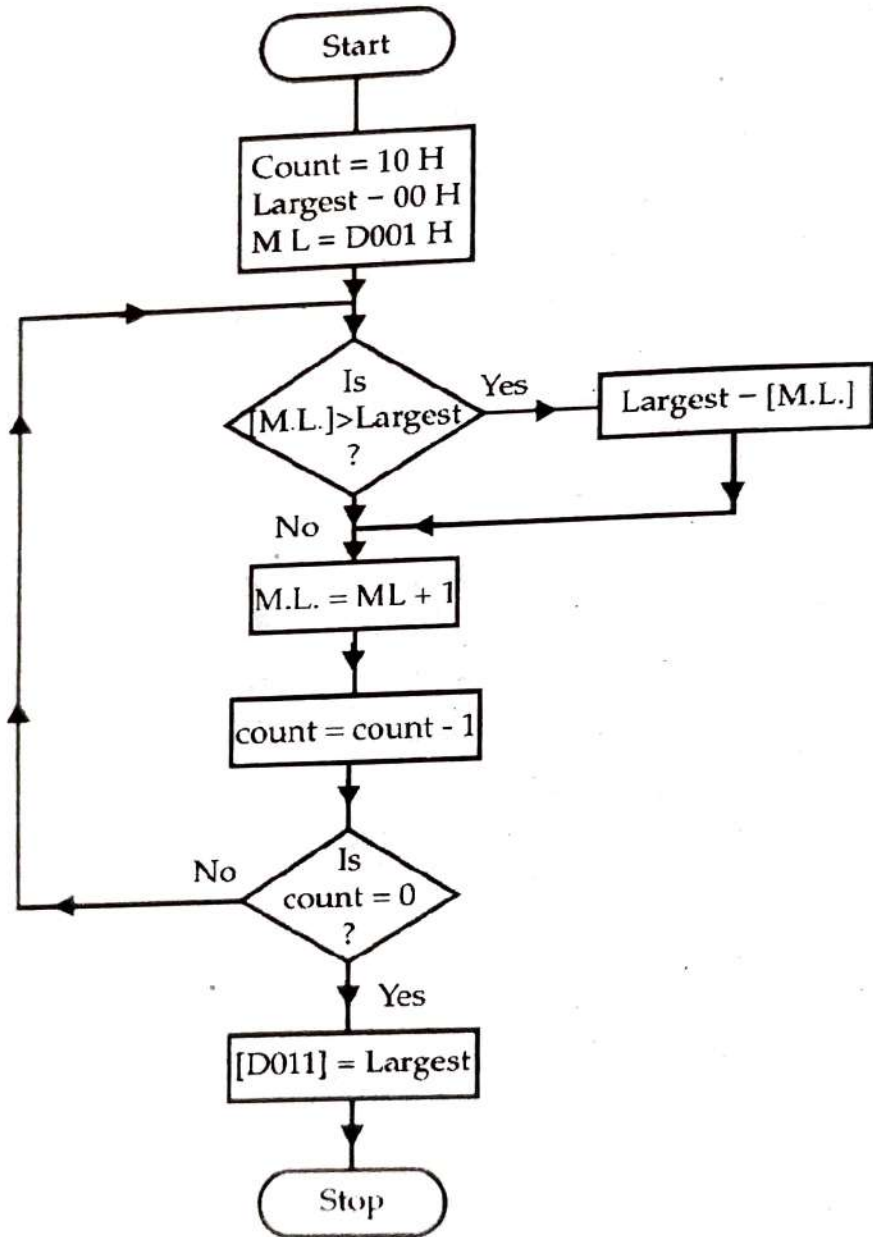
**Welcome to tpspoint**  
**Assembly Language Programing ( PART 1)**

**All lessons of Computer Science 1& 2 pdf are given in the website [www.tpspoint.com](http://www.tpspoint.com)**

Assembly Language Programs

- 1) A series of numbers are stored in memory from D001 H to D010 H. Write a program in assembly language to find largest number among these numbers. Store the largest number in memory location D011H.

Flow chart :



## Assembly language program :

Memory address	Label	Mnemonics	Comments
C000		LXI H, D001H	; Set H-L pair to D001H
C003		MVI A, 00H	; Largest = 00 H
C005		MVI C, 10H	; Set Count = 10 H
C007	Loop	CMP M	; Compare with previous no. Is it greater ?
C008		JNC AHEAD	; No larger is in Acc. Go to AHEAD.
C00B		MOV A, M	; get larger no. in acc
C00C	AHEAD	INX H	; Address of next memory
C00D		DCR C	; Count = Count - 1
C00E		JNZ Loop	; Repeat if count $\neq$ 0
C011		MOV M A	; Store largest no in M. L. D011
C012		HLT	; Stop

Note : To find smallest number, initially set smallest = FFH.

i.e. change instruction

C003 MVI A, FFH; smallest = FFH

And set smallest = [M.L.] if smallest > [M.L.]

i.e. change instruction

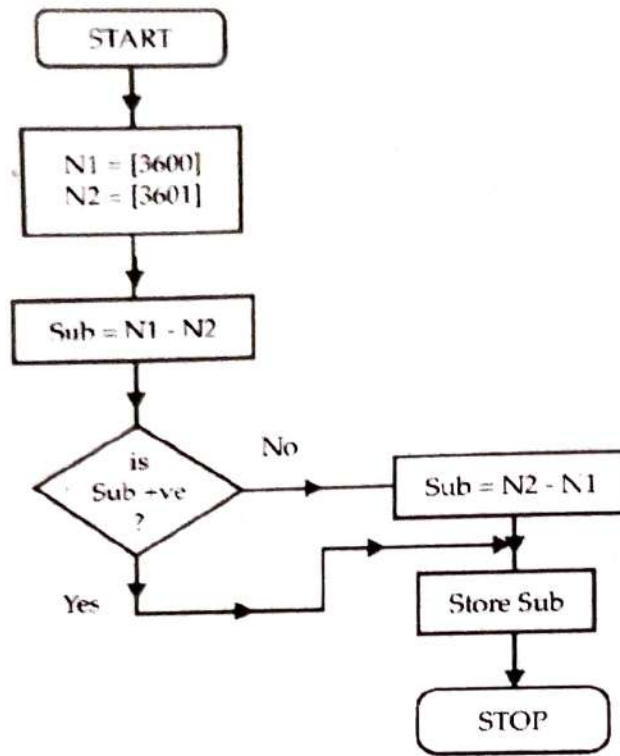
C008 JC AHEAD; Yes, smallest is in acc.

- 2) Write an assembly language program to subtract the number stored in memory location 3601 from the number stored in memory location 3600 H. Store the positive result at location 3602 H. **(Mar. 02)**

## Assembly language program

Memory address	Label	Mnemonics	Comments
C000		LXI H, 3600H	; Set H-L pointer to 3600H
C003		MOV A, M	; Take 1 <sup>st</sup> no. in Acc
C004		INX H	; Increment H L contents
C005		SUB M	; Subtract II <sup>nd</sup> no. from I <sup>st</sup> no.
C006		JP escape	; escape if result is positive
C009		MOV AM	; If result is negative then
C00A		DCX H	Subtract I <sup>st</sup> no. from II <sup>nd</sup>
C00B		SUB M	number
C00C	escape	STA 3602 H	; Store the +ve result at memory location 3602H
C00F		HLT	; Stop

**Flowchart :**

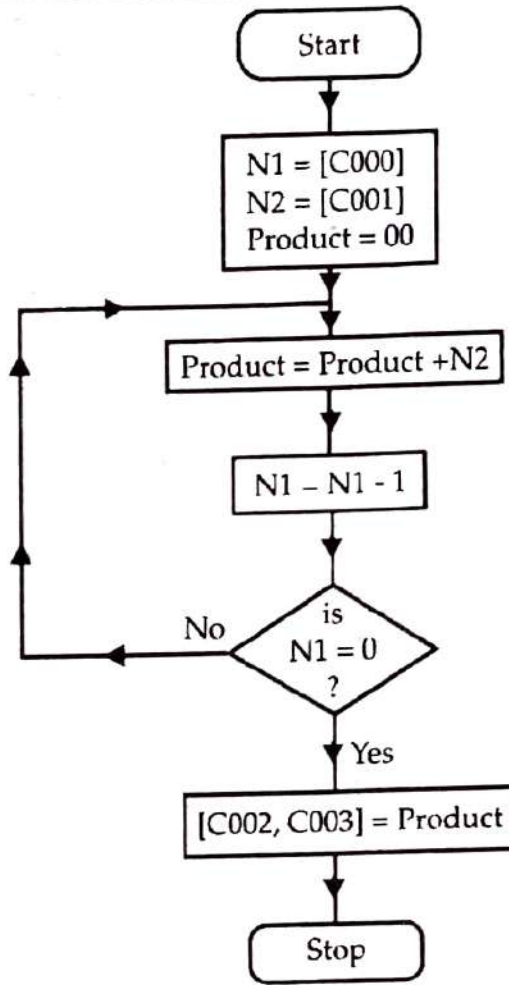


- 3) Write a program in assembly language that multiply two 8-bit hex numbers stored in memory locations C005H and C006H. Store the two byte result in consecutive memory locations starting from C000H. (March 2003, 2005)

**Assembly language program :**

Memory address	Label	Mnemonics	Comments
D000		LXI H, 0000H	; Set initial product = 0
D003		LDA C005 H	; Set [Acc] = N1
D006		MOV E, A	; Set [E] = N1
D007		LDA C006 H	; Set [Acc] = N2
D00A		MVI D, 00H	; Set [D] = 00H
D00C	Loop	DADD	; product = product + N1
D00D		DCR A	; N2 = N2 - 1
D00E		JNZ Loop	; Repeat, if N <sub>2</sub> ≠ 0
D011		SHLD C000 H	; Store product in C000 and C001
D014		HLT	; Stop

Flowchart :



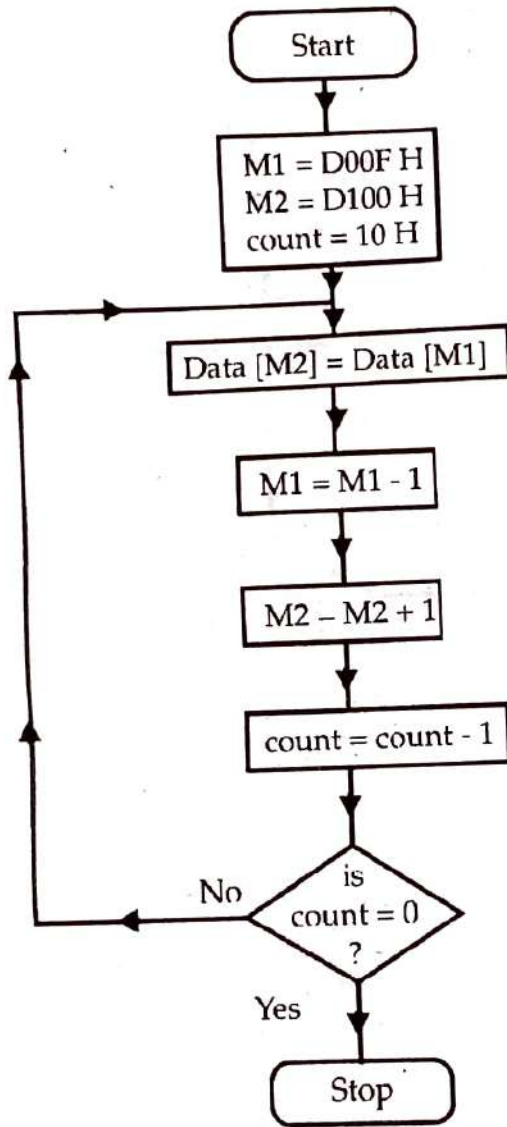
- 4) A block of data is stored in memory from D000H to D00F H. Write a program to shift the data contents of the block in reverse order, starting from memory location D100H.

Assembly language program :

Memory address	Label	Mnemonics	Comments
C000	Loop	LXI H, D00F H	; set up HL as a pointer to source
C003		LXI D, D100 H	; set up DE as a pointer to destination
C006		MVI B, 10 H	; set up B to count 16 bytes
C008		MOV A, M	; get data byte from memory
C009		STAX D	; Store data byte at destination
C00A		DCX H	; Decrement source pointer
C00B		INX D	; Increment destination pointer
C00C		DCR B	; Decrement count
C00D		JNZ Loop	; if not zero, go back
			C010 HLT



Flowchart :

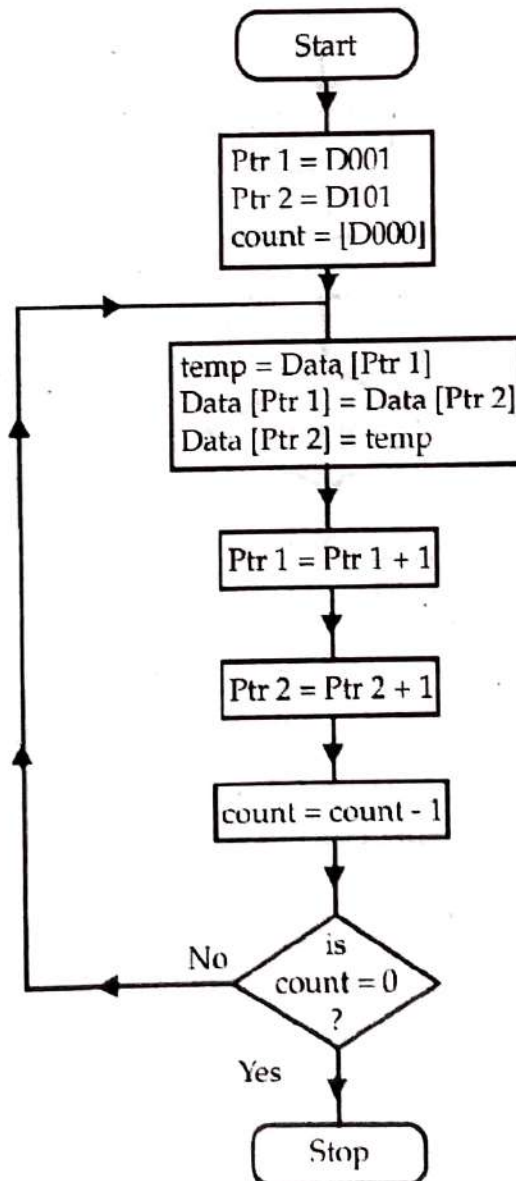


5) A block of data is stored in memory from D001H. The length of block is stored in D000H. Another block of same length is stored from D101H. Write a program in assembly language to exchange the contents of these two blocks. (March 2005)

Memory address	Label	Mnemonics	Comments
C000		LXI H, D000 H	; Set up HL as source memory
C003		LXI D, D101 H	; Setup DE as an index for destination
C006		MOV B, M	; Setup B to count = [D000]
C007	NEXT	INX H	; Pointer to next source location
C008		MOV C, M	; Get databyte from source memory

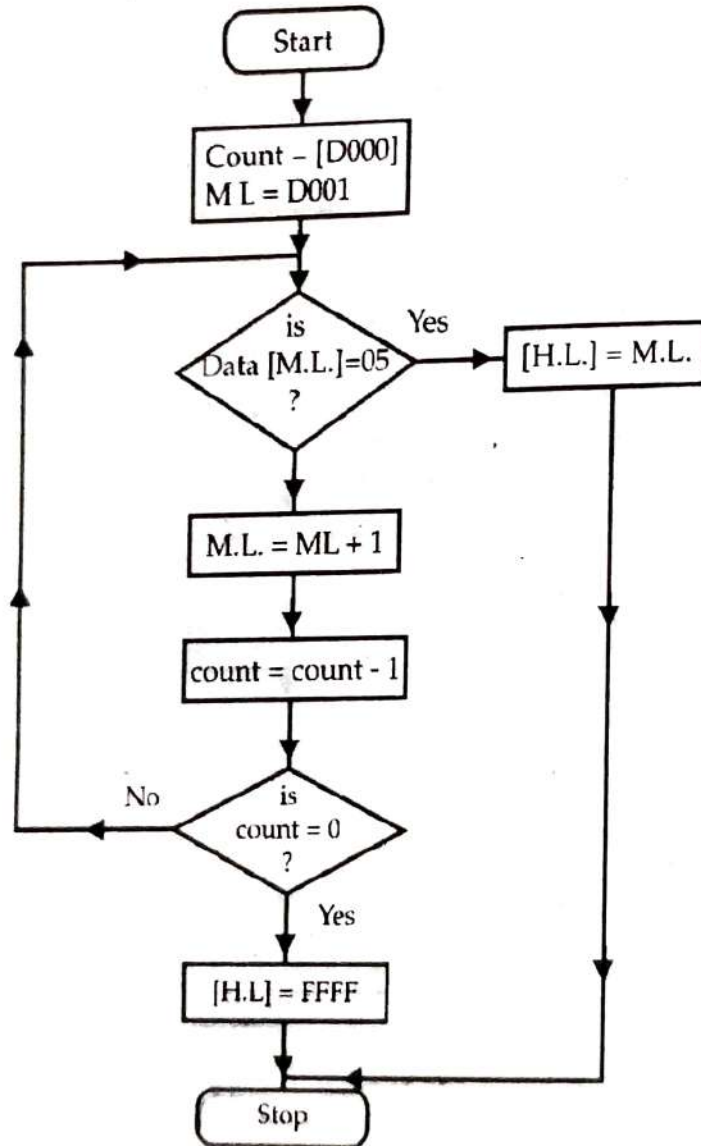
Memory address	Label	Mnemonics	Comments
C009		LDAX D	; Load databyte from destination in acc.
C00A		MOV M, A	; Store [A] in first block
C00B		MOV A, C	; Copy [C] into [A]
C00C		STAX D	; Store [A] at second block
C00D		INX D	; Pointer to next destination location
C00E		DCR B	; decrement count
C00F		JNZ NEXT	; Repeat the loop
C012		HLT	; Stop

Flowchart :



- 6) A block of data is stored in memory from D001. The length of block is stored in memory location D000H. Write a program that searches for first occurrence of data 05 H in given block. Store the address of this occurrence in H-L pair. If the number is not found, then H-L pair should contain FFFF H.

Flow chart :



Assembly language program :

Memory address	Label	Mnemonics	Comments
C000		LXI H, D000H	; Set H-L pair to D000H
C003		MOV C, M	; Set count = [D000]
C004		MVI A, 05H	; Set [Acc.] = 05 H
C006	Loop	INX H	; [H-L] = [H-L] + 1



Memory address	Label	Mnemonics	Comments
C007		CMP M	; Is [M-L] = 05 ?
C008		JZ escape	; escape, if [M-L] = 05 H
C00B		DCR C	; count = count - 1
C00C		JNZ Loop	; repeat, if count $\neq$ 0
C00F		LXI H, FFFF H	; Set H-L pair to FFFF if number is not found
C012	escape	HLT	; Stop

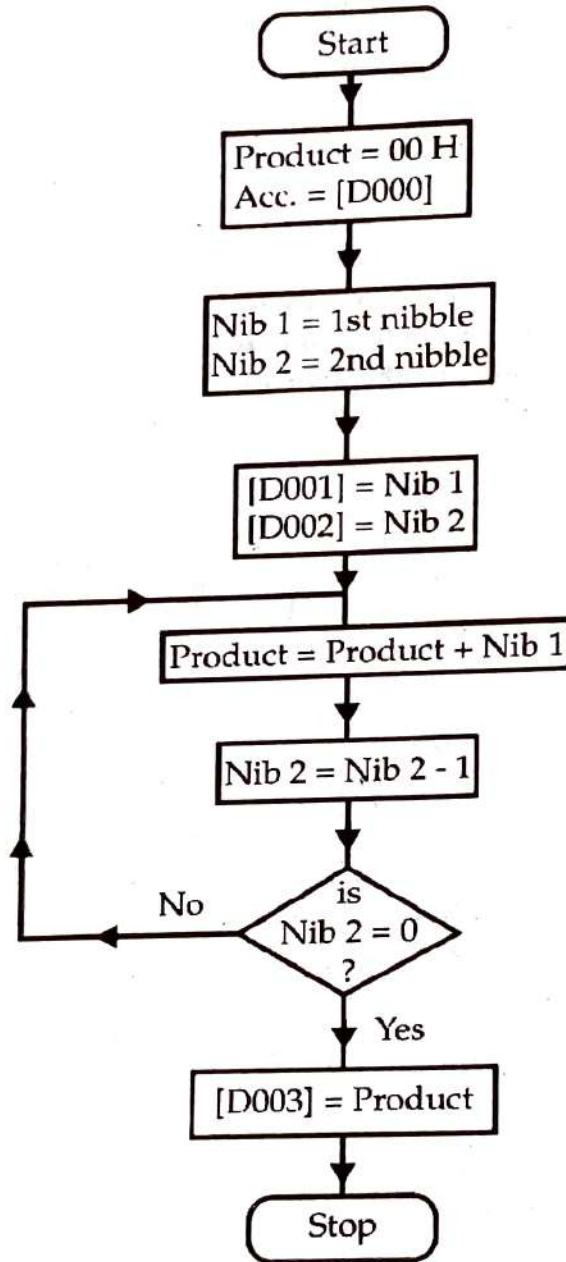
- 7) Write an ALP that separates the two nibbles of an 8-bit hex number stored in memory location D000H. Store the same in memory locations D001 and D002H. The program must also multiply the two nibbles and store the product in memory location D003H.

Assembly language program :

March 2011

Memory address	Label	Mnemonics	Comments
C000		LXI H, D000H	; Set H-L pointer to D000H
C003		MOV A, M	; take no. in Acc.
C004		ANI 0F H	; separate the first nibble
C006		MOV B, A	
C007		MOV A, M	; Again take no. in Acc.
C008		ANI F0H	; separate 2nd nibble
C00A		RRC	; with four rotate
C00B		RRC	; instructions make
C00C		RRC	; 4 MSB bits to 4 LSBS
C00D		RRC	
C00E		MOV C, A	; store in register C
C00F		INX H	; store first nibble in D001H
C010		MOV M, B	
C011		INX H	; store second nibble in D002H
C012		MOV M, C	
C013		SUB A	; initial product = 0
C014	Loop	ADD B	; product = product + Nib 1
C015		DCR C	; Nib2 = Nib2 - 1
C016		JNZ Loop	; repeat if Nib2 $\neq$ 0
C019		INX H	; store product in D003 H
C01A		MOV M, A	
C01B		HLT	; Stop

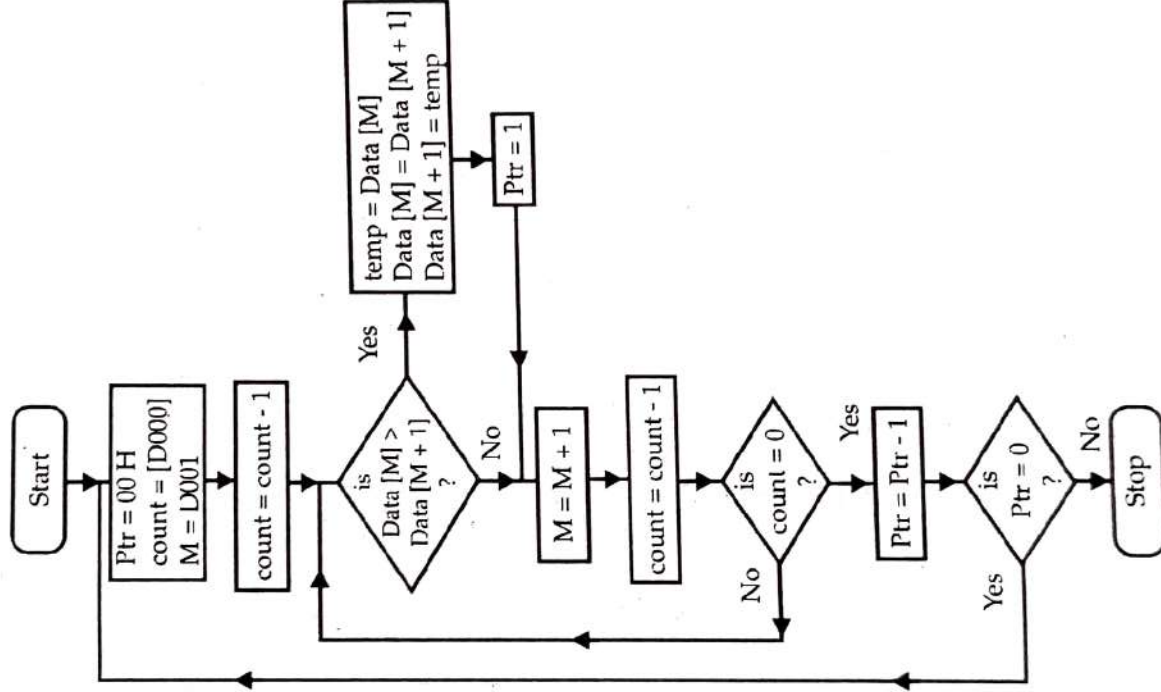
**Flowchart :**



8)

A block of data is stored in memory starting from memory location D001H. The length of block is stored at memory location D000H. Write a program in assembly language to sort the content of block in ascending order.

## Flowchart :



**Assembly language program :**

Memory address	Label	Mnemonics	Comments
C000	Start	MVI B, 00H	; Set ptr = 00H initially
C002		LXI H, D000	; Set H-L pair to D000H
C005		MOV C, M	; Set count = [D000]
C006	Loop	INX H	; Increment HL reg pair by 1
C007		MOV A, M	; take first no. in acc.
C008		DCR C	; count = count - 1
C009		INX H	; [H-L] = [H-L] + 1
C00A		CMP M	; Is [Acc.] > [[H-L]] ?
C00B		JC escape	; Go to escape if [Acc.] < [[H-L]]
C00E		MOV D, M	; move mem content to reg. D
C00F		MOV M, A	; Interchange two) number if [Acc.] > [[H-L]]
C010		DCX H	
C011		MOV M, D	; move reg. D content to mem
C012	INX H	; Increment HL reg pair by 1	
C013	MVI B, 01H	; Set ptr = 01 H	
C015	escape	DCR C	; count = count - 1
C016		JNZ Loop	; repeat if count $\neq$ 0
C019		DCR B	; ptr = ptr - 1
C01A		JZ Start	; go to start, if ptr = 0
C01D		HLT	; Stop

**Note :** For descending order, only the change is to use JNC escape in place of JC escape in above program.

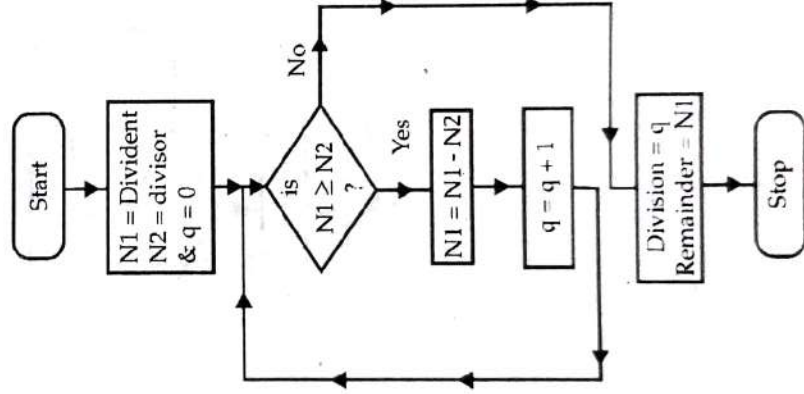
- 9) Write an assembly language program that divides two one byte Hex numbers where dividend is stored in memory location D000H and divisor is stored in memory location D001H. Store quotient and remainder in memory locations D002H and D003H respectively.

**Assembly language program :**

Memory address	Label	Mnemonics	Comments
C000	Loop	LXI HD000 H	; Set H-L pointer to dividend
C003		MVI C, 00H	; Set Initial quotient = 00
C005		MOV A, M	; Set Acc. = Divident
C006		INX H	; Set H-L pointer to divisor
C007		CMP M	; Is $N1 \geq N2$ ?

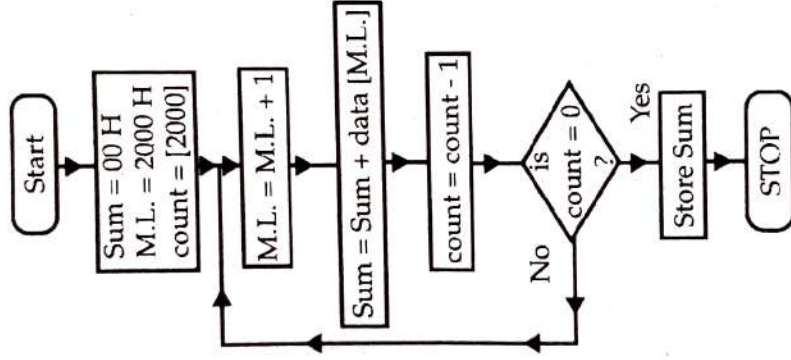
Memory address	Label	Mnemonics	Comments
C008		JC Escape	; Go to Escape if $N1 < N2$
C00B		SUB M	; $N1 = N1 - N2$
C00C		INR C	; quotient = quotient + 1
C00D		JMP Loop	; jump to again compare N1 and N2
C010	Escape	INX H	; Increment HL reg pair by 1
C011		MOV M, C	; Store C reg content i.e. quotient in D002 H
C012		INX H	; Increment HL reg pair by 1
C013		MOV M, A	; Store acc content i.e. remainder in D003 H
C014		HLT	; Stop

**Flowchart :**





- 10) A block of data is stored from location 2001H and onwards. The count is stored at location 2000H. Write an assembly language program to find out sum of the data items stored in block. Store the result at location 2500H and onwards starting with L.S.B. (Mar. 02)



Assembly language program :

Memory address	Label	Mnemonics	Comments
C000		LXI H, 2000 H	; Set HL pointer to 2000H
C003		MOV C, M	; Get count in register C
C004		MVI A, 00H	; Make LSBs of sum = 00
C006		MOV B, A	; Make MSBs of sum = 00
C007	LOOP	INX H	; Set HL to point num in series
C008		ADD M	; Previous No. + Next No.

Memory address	Label	Mnemonics	Comments
C009		JNC AHEAD	; Is carry ? No, goto AHEAD
C00B		INR B	; Yes, add carry to MSBs of sum
C00C	AHEAD	DCR C	; Decrement count
C00D		JNZ LOOP	; Is count = 0 ? No, jump to loop
C010		STA 2500H	; Store LSBs of the sum to 2500 H
C013		MOV A, B	; Get MSBs of sum in accumulator
C014		STA 2501 H	; Store MSBs
C017		HLT	; Stop

- 11) Write an assembly language program to add two 8 bit BCD numbers stored at memory locations 5000H and 5001H. Store the result at memory location 5002 H onwards starting with least significant bit. **(Mar-02, Oct-03)**

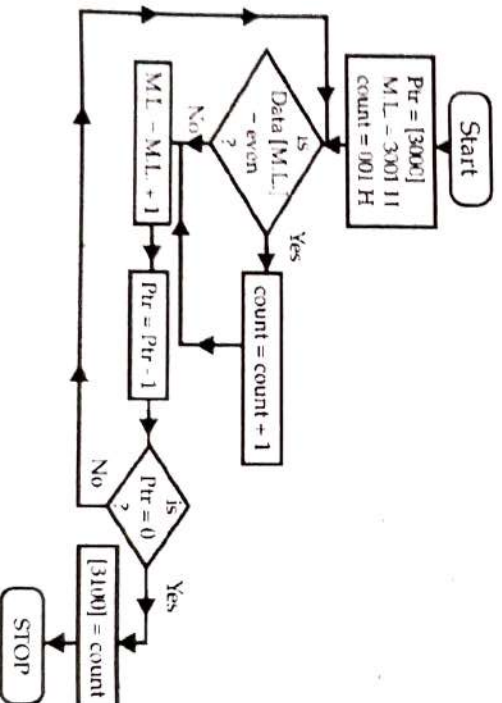
Ans. :

Memory address	Label	Mnemonics	Comments
C000		LXI H, 5000 H	; Initialize H-L pair with address of first number
C003		MVI C, 00H	; Initialize register C to store MSB.
C005		MOV A, M	; Get first number in accumulator.
C006		INX H	; Address of next number in H-L pair
C007		ADD M	; Add two numbers
C008		DAA	; Decimal adjust accumulator.
C009		JNC L1	; Jump if no carry to label L1
C00C		INR C	; If carry, increment MSB in register C.
C00D	L1	STA 5002H	; Store the LSB of SUM in location 5002H
C010		MOV A, C	; Get MSB in accumulator
C011		STA 5003H	; Store the MSB of SUM in location 5003H
C014		HLT	; Stop the processing

- 12) Write an assembly language program to count number of even data bytes occurring in a block stored from memory location 3001H and onwards. The length of block is stored in location 3000H. Store the result in location 3100H. **(Mar-02)**

**Assembly language program :**

Memory address	Label	Mnemonics	Comments
C000		LXI H, 3000 H	; Initialize HL pointer to 3000H
C003		MVI B, 00H	; Initialize register B to store count
C005		MOV C, M	; Get length of block in C
C006	Loop	INX H	; Increment H-L pair by 1
C007		MOV A, M	; Get number in accumulator
C008		RRC	; Check even number
C009		JC AHEAD	; Jump on carry i.e. if no. is odd
C00C		INR B	; No carry - increment count
C00D	AHEAD	DCR C	; Decrement C by 1
C00E		JNZ Loop	; Is zero ? No - jump to Loop
C011		MOV A, B	; Store count in accumulator
C012		STA 3100H	; Store result in 3100 H
C015		HLT	; Stop

**Flowchart :**

- 13) A hex number is stored at location 3000 H. Write an assembly language program to interchange its digits. The new number is to be stored at 3001. Add original number with new number and store the result at location 3010 H. **March 2002, Oct. 2010**

Algorithm :

Step 1 : Set : [Acc.] = [3000]

Step 2 : Repeat For I = 1 To 4

Rotate [Acc.] one bit right.

Step 3 : Set : [3001] := New no.

Step 4 : Sum = New no. + original no.

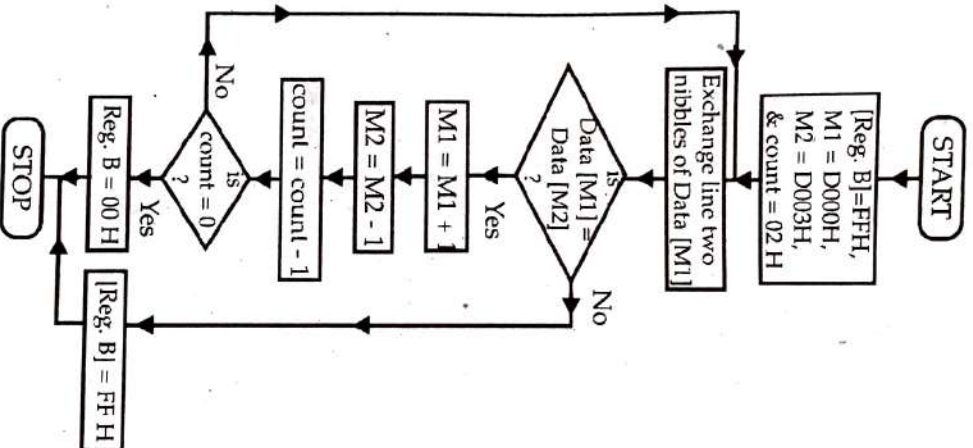
Step 5 : Set : [3002] = Sum

Step 6 : Exit.

Assembly language program :

Memory address	Label	Mnemonics	Comments
C000		LXI, 3000 H	; Set H-L pointer to 3000H
C003		MOV A, M	; take no. in accumulator
C004		RRC	; with 4 RRC instructions
C005		RRC	; Interchange the digits of the no
C006		RRC	
C007		RRC	
C008		INX H	
C009		MOV M, A	Increment HL reg pair by 1
C00A		DCX H	; store the exchanged no. in 3001 H
C00B		ADD M	; Decrement HL reg pair by 1
C00C		STA 3010H	; Add new no. & original no.
C00F		HLT	; Store result in 3010H ; Stop

- 14) A 4-byte hex number, beginning with lower order byte is stored from memory location D000H. Write an ALP that checks whether the given number is palindrome or not. If the number is palindrome, then register B contains 00H. Else, it contains FFH.

**Flowchart :****Assembly language program :**

Memory address	Label	Mnemonics	Comments
C000		LXI H, D000 H	; Set H-L pair to D000H
C003		LXI D, D003H	; Set D-E pair to D003H
C006		MVI C, 02H	; Set count = 02H

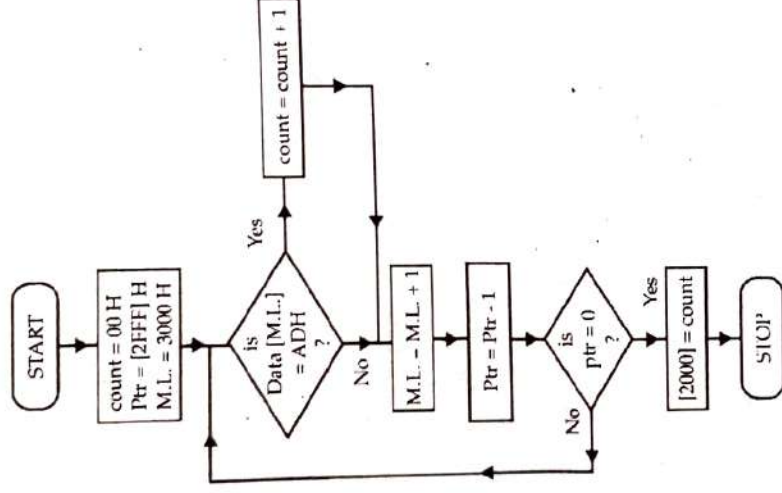


Memory address	Label	Mnemonics	Comments
C008	Loop	MOV A, M	; take [[H-L]] in accumulator
C009		RRC	; with 4 RRC
C01A		RRC	; exchange the
C00B		RRC	; two nibbles of
C00C		RRC	; no. in accumulator
C00D		XCHG	; Exchange [H-L] & [D-E]
C00E		CMF M	; is [Acc.] = [[H-L]] ?
C00F		JNZ Escape	; escape if [Acc.] ≠ [[H-L]]
C012		XCHG	; Reexchange [HL] & [DE] pairs
C013		INX H	; [H-L] = [H-L] + 1
C014		DCX D	; [D-E] = [D-E] - 1
C015		DCR C	; count = count - 1
C016		JNZ Loop	; Repeat if count ≠ 0
C019		MVI B, 00H	; Set [Reg. B] = 00
C01B		IMP STP	if No. is palindrome.
C01E	Escape	MVI B, FFH	; Number is not palindrome
C020	STP	HLT	; stop

- 15) Write an assembly language program to count the number of times the data ADH is found in a block of memory locations starting from 3000H. Length of block is stored in location 2FFFH. Store the result in location 2000H. **[March 2002]**

Assembly language program :

Memory address	Label	Mnemonics	Comments
C000		MVI B 00H	; Set Count = 00 H
C002		LXI H, 2FFF H	; Set H-L pointer to 2FFF H
C005		MOV C, M	; Get count in register C
C006	Loop	INX H	; [H-L] = [H-L] + 1
C007		MOV A, M	; Check whether
C008		CPI ADH	; [[H-L]] = ADH or not.
C00A		JNZ NEXT	; If zero ? No-jump to NEXT
C00D		INR B	; count = count + 1
C00E	NEXT	DCR C	; Decrement count
C00F		JNZ Loop	; Repeat loop if count ≠ 0
C012		MOV A, B	; Yes, store count in A
C010		STA 2000 H	; Store count in 2000 H
C016		HLT	; Stop

**Flowchart :**

16) Write a program to complement each flag in the flag register.

Memory address	Label	Mnemonics	Comments
C000		PUSH PSW	; Save flags on stack
C001		POP H	; Retrieves flags in 'L'
C002		MOV A, L	; Flags in accumulator
C003		CMA	; Complement accumulator
C004		MOV L, A	; Accumulator in Reg 'L'
C005		PUSH H	; Save on stack
C006		POP PSW	; Back to flag register
C007		HLT	; Stop

17) Write a subroutine to fill the memory locations 2800H to 28FFH with Hex numbers 00H to FFH respectively.

Memory address	Label	Mnemonics	Comments
F000	START	LXI D, 2800 H	; Set memory start address
F003		XRA A	; Clear acc. and carry
F004		MVI B FFH	; load counter
F006	LOOP	STAX D	; Store data in memory
F007		INR A	; increment data
F008		INR E	; increment memory address
F009		DCR B	; count = count - 1
F00A		JNZ LOOP	; go back if not over
		F00D RET	; return to main program if over

18) Trace the following program and fill in the blanks.

```

MVI B, 08 H
MVI C, 03 H
MVI A, 01 H
ANI 05 H
STA D000 H
ADD C
MOV D, C

```

**Result :**

(i) [Reg. A] = ...  
(ii) [Reg. B] = ...  
(iii) [Reg. C] = ...  
(iv) [Reg. D] = ...  
(v) [D000] = ...

Ans. : Given program is :

Mnemonics	Comments
MVI B, 08 H	: Set reg. B to immediate data 08 H i.e. [reg. B] = 08 H
MVI C, 03 H	: Set reg. C to immediate data 03 H i.e. [Reg. C] = 03 H
MVI A, 01 H	: Set accumulator to immediate data 01 H i.e. [Acc.] = 01 H
ANI 05 H	: Logically AND data 05 H with [A] i.e. [A] : 00000001 AND 05 : 0000101 00000001 = 01 i.e. [A] = 01 H.
STA D000 H	: Move [Acc.] m. l. [D000] i.e. [D000] = 01 H.
ADD C	: Add [C] to [Acc.] and store result in Acc. [Acc.] = 01 H = 00000001 + [C] = 03 H = 00000011 00000100 i.e. [Acc.] = 04 H
MOV D, C	: Move contents of reg. C to reg. D i.e. [reg. D] = 03 H

**Final Result :**

- (i) [Reg. A] = 04 H                      (ii) [Reg. B] = 08 H                      (iii) [Reg. C] = 03 H  
(iv) [Reg. D] = 03 H                      (v) [D000] = 01 H

- 19) Write an assembly language program to copy a block of data having starting address 8900 H to the new location starting from 9100H. The length of block is stored at memory location 8FFFH.

October 2003

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI H 8FFFH LXI B, 9100 H	; Initialize H-L pair with address of count ; Initialize B-C pair with starting address of destination block
LOOP	MOV D, M INX H MOV A, M STAX B INX H INX B DCR D JNZ LOOP END HLT	; Copy the count in Register D. ; H-L pair points to the starting address of source block ; Transfer the memory content to accumulator. ; Store the accumulator content to new location. ; Increment H-L pair ; Increment B-C pair ; Decrement count ; Jump if no zero to label LOOP ; Stop processing.

- 20) Write an assembly language program to add two 8 bit BCD numbers stored at memory location 5000H and 5001H. Store the result at memory location 5002 H onwards starting with least significant bit.

October 2003

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI H, 5000H  MVI C, 00H MOV A, M INX H ADD M DAA JNC L1 INR C	; Initialize H-L pair with address of first number ; Initialize register C to store MSB. ; Get first number in accumulator. ; Address of next number in H-L pair ; Add two numbers. ; Decimal Adjust accumulator ; Jump if no carry to label L1 ; If carry increment MSB in register C.
L1	STA 5002H MOV A, C STA 5003 H END HLT	; Store the LSB of SUM in location 5002 H ; Get MSB in accumulator ; Store the MSB of SUM in location 5003 H ; Stop the processing



- 21) Write an assembly language program to find out 2's complement of five numbers stored from memory location 3330H and onwards. Store the result from memory location address 4100H. **(Oct. 03)**

Ans. :

Label	Mnemonics+ Operand	Comments
START	LXI H, 3330H	; Initialize H-L pair with address of first number.
	LXI B, 4100H	; Initialize B-C pair with destination address.
	MVI D, 05H	; Store count in register D
LOOP	MOV A, M	; Get the number in accumulator
	CMA	; 1's complement of number in accumulator
	INR A	; 2's complement of number in accumulator.
	STAX B	; Store 2's complement at address pointed to by BC pair.
	INX H	; Increment H-L pair
	INX B	; Increment B-C pair
	DCR D	; Decrement count in register D
	JNZ LOOP	; Is count zero ? no-jump to lable LOOP
END	HLT	; Stop the processing

- 22) A block of data is stored in memory location from 9107H to 91FFH. Write an assembly language program to transfer the block in reverse order to memory location 9200H and onwards. **(Oct. 03, Oct. 2008; July 18)**

Ans. :

Label	Mnemonics+ Operand	Comments
START	LXI H, 91FF H	; Initialize H-L pair with address for last location of source block
	LXI B, 9200 H	; Initialize B-C pair with destination address.
	MOV D,L	; Get the count FFH in register D.
LOOP	MOV A,M	; Get the number in accumulator.
	STAX B	; Store the number at address pointed to by B-C pair.
	DCX H	; Decrement H-L pair
	INX B	; Increment B-C pair
	DCR D	; Decrement count
	JNZ LOOP	; Is count zero ? No - jump to label LOOP
END	HLT	; Stop processing.



- 23) Write an assembly language program to count the number of odd data bytes occurring in a block starting from the memory location address 7501H to 75FFH Store the result at the memory location 7600H.

Ans. :

(Mar. 2005, Oct. 2003)

Label	Mnemonics+ Operand	Comments
START	LXI H, 7501 H MVI C, FFH MVI B, 00H MOV A, M RRC	; Initialize H-L pair with starting address ; Store count FFH in register C. ; Initialize odd count to zero in register B ; Get the number in accumulator ; Rotate to determine odd or not ? ; Is carry ? No-jump to label GO.
GO	JNC GO INR B INX H DCR C	; Yes - Increment odd count ; Increment H-L pair ; Decrement count
END	JNZ LOOP MOV A, B STA 7600 H HLT	; Is count zero ? No - jump to lable LOOP ; Get odd count in accumulator ; Store odd count at 7600 H ; Stop processing.

- 24) Write an assembly language program to perform the multiplication of two 8-bit numbers where multiplicand is stored at the memory locations 2501H and 2502H and multiplier is stored at 2503H. The result is to be stored at memory location address 2504H and 2505H.

(Note : 8 bit multiplicand is extended to 16 bit)

Ans. :

(Oct. 2003)

Label	Mnemonics+ Operand	Comments
START	LHLD 2501 H XCHG LDA 2503 H LXI H, 0000H MVI C, 08H	; Get multiplicand in H-L pair ; Multicand in D-E pair ; Multiplier in accumulator ; Initial value of product equal to 0000H in H-L pair ; Count is 8 in register C
LOOP	DAD H RAL JNC GO	; Shift partial product left by one bit ; Rotate multiplier left by one bit. ; Is multiplier bit = 1 ?
GO	DAD D DCR C JNZ LOOP SHLD 2504 H HLT	NO - goto label GO ; Product = Product + Multicand ; Decrement count ; jump if no zero to label LOOP ; store the result ; stop processing

- 25) Write an assembly language program to divide a hexadecimal number stored in a memory location 8000 H by a hexadecimal number stored in memory location 8001 H. Store the quotient at 8002 H and remainder at 8003 H. (March, 2004, Oct-2004)

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI H, 8000 H MVI C, 00H MOV A, M INX H CMP M JC ESCAPE SUB M INR C JMP LOOP INX H MOV M, C INX H MOV M, A HLT	; Initialize H-L pair with dividend address ; Initialize quotient to 00H in register C. ; Get the dividend in accumulator ; Set H-L pointer to divisor. ; Is dividend $\leq$ divisor ? ; If yes, go to ESCAPE ; If no, subtract divisor from dividend ; Increment quotient by 1 ; Jump to again LOOP ; Increment H-L pair ; Store quotient in 8002 H ; Increment H-L pair ; Store remainder in 8003 H ; Stop processing.
LOOP		
ESCAPE		
END		

- 26) An 8-bit number is stored in memory location C400H. Write an assembly language program to count the 'zero' in the given number. Store the count in memory location C500H. (March, 2004)

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI H, C400H MOV B, M MVI C, 00H MVI E, 08H MOV A, B RLC MOV B, A JC DOWN INR C DCRE JNZ Loop MOV A, C STA C500H HLT	; Initialize H-L pair with address of number ; Get number in B register ; Initialize register C to store count of zeros. ; Initialize register E to store counter for 8-bit number. ; Transfer the 8-bit number into accumulator ; Rotate content of accumulator left side by one bit to check the bit. ; Store the rotated data in register B. ; If carry? Yes, jump to DOWN ; Increment register C contents by 1 if there is no carry means bit is zero ; Get answer i.e. number of zeros to accumulator ; Jump if no zero to Loop ; Get answer i.e. number of zero's to acc. ; Store the count in location C500H ; Stop the processing
Loop		
DOWN		
END		

- 27) Write an assembly language program to transfer first 10 bytes of memory block starting from 5000 H to a new block starting from 5020 H. **(March, 2004)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI H, 5000H	; Initialize H-L pair with starting address of source block.
	LXI B, 5020 H	; Initialize B-C pair with starting address of destination block.
	MVI D, 0A H	; Initialize register D to store the count 0AH
LOOP	MOV A, M	; Transfer the memory content to accumulator
	STAX B	; Store the accumulator content to new location
	INX H	; Increment H-L pair
	INX B	; Increment B-C pair
	DCR D	; Decrement count
	JNZ LOOP	; Jump if no zero to label LOOP
END	HLT	; Stop the processing

- 28) Write an assembly language program to generate the Fibonacci's series for first eight numbers. Store the series in a memory block starting from C100H.

(Note : The first hexanumbers of series are 00, 01, 01, 02, 03, 05, 08, 0D) **(March, 2004)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI H, C100H	; Initialize H-L pair with starting address
	MVI D, 06 H	; Store count 06H in register D
	MVI B, 00H	; Initialize register B with first term of series i.e. 00H
	MOV M, B	; Copy the first term at memory
	INX H	; Increment H-L pair
	MVI C, 01H	; Initialize register C with second term of series 01 H
	MOV M, C	; Copy second term at memory
LOOP	INX H	; Increment H-L pair
	MOV A, B	; Get the number in accumulator
	ADD C	; Generate next term by adding previous two terms.
	MOV M, A	; Copy the result at memory location
	MOV B, C	; Copy the content of register C into B register
	MOV C, A	; Copy the content of accumulator into C register
	DCR D	; Decrement count
	JNZ LOOP	; Is count zero ? No-jump to label LOOP
END	HLT	; Stop processing



- 29) The two BCD numbers are stored at 3400H and 3401H. Write an assembly language program to add these BCD numbers and store the result in memory locations 3402 H and 3403 H. (March-04)

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI H, 3400 H	; Initialize H-L pair with address of first BCD number.
	MVI B, 00H	; Initialize register B to store MSB of first number
	MVI D, 00H	; Initialize register D to store MSB of second number
	MOV C, M	; Get first number into register C.
	INX H	; Increment H-L pair to point at second number.
	MOV E, M	; Get second number into register E.
	XCHG	; Exchange the contents of H-L pair with D-E pair
	DAD B	; Add first number with second number.
	XCHG	; Exchange the contents of H-L pair with D-E pair
	MOV A, E	; Transfer contents of register E to accumulator
	DAA	; Decimal adjust lower order byte of sum
	INX H	; Increment H-L pair
	MOV M, A	; Store lower order byte of sum to memory location 3402 H
	MOV A, D	; Transfer contents of register D to accumulator
	DAA	; Decimal adjust higher order byte of sum.
	INX H	; Increment H-L pair
	MOV M, A	; Store higher order byte of sum of 3403 H.
END	HLT	; Stop the processing

- 30) Write a assembly language program to count the occurrence of the data 9CH in a memory block starting from 4000H to 400FH. Store the count at memory location 4500H. (Mar-2004)

Ans.:

Label	Mnemonics+ Operand	Comments
START	MVI C, 10 H	; Store count 10 H (16 bytes) in register C
	MVI B, 00H	; Initialize occurrence count in register B.
	LXI H, 4000H	; Initialize H-L pair with starting address
LOOP	MOV A, M	; Get the number in accumulator
	CPI 9CH	; Check whether the number in accumulator is 9CH
	JNZ NEXT	; If no ? jump to label NEXT
	INR B	; Yes, increment content in register B by 1.
NEXT	INX H	; Increment H-L pair
	DCR C	; Decrement count
	JNZ LOOP	; Is count zero ? No-jump to label LOOP
	MOV A, B	; Get count in accumulator
	STA 4500, H	; Store count of occurrence at 4500 H
END	HLT	; Stop processing

- 31) Two three - byte numbers are stored in BCD and EHL registers. Write an assembly language program to find their sum and store the result in EHL. **(Oct.03)**

Ans. :

Label	Mnemonics+ Operand	Comments
START	MOV A, L	; Get least significant byte in A
	ADD D	; Add with D without carry
	MOV L, A	; Result in L
	MOV A,H	; Get middle byte in A
	ADC C	; Add with carry
	MOV H, A	; Result in H
	MOV A,E	; Get most significant byte in A
	ADC B	; Add with carry
	MOV E, A	; Result in E
	HLT	; stop processing

- 32) Write an assembly language program to divide data at location 1050 by data stored at location 1051. Store the quotient and remainder in 1060 and 1061 memory locations respectively.

Ans. : (Please . see similar Question 25 page No - 2 - 81 & Make Necessary changes) **(Oct.04)**

- 33) The length of block is in memory location 1070 and block itself begins from 1071. Write a program in assembly language to store the count of odd numbers in register C. **(Oct.04)**

Ans. :

Label	Mnemonics+ Operand	Comments
START	LXI H, 1070 H	; Initialize H-L pair with starting address
	MOV D, M	; Store length of block in D
	INX H	; Point to next memory location
	MVI C, 00H	; Initialize odd count to zero
LOOP	MOV A, M	; Get the number in accumulator
	RRC	; Rotate to determine odd or not ?
	JNC GO	; Is carry ? NO jump to label GO
	INR C	; Yes - Increment odd count
GO	INX H	; Increment H-L pair
	DCR D	; Decrement count
	JNZ LOOP	; Is count zero ? No-jump to label loop
	HLT	; stop processing



- 34) Write a program in assembly language to transfer a block of data from 1050 to 1059 to memory location whose starting address is 1070 using exchange (XCHG) instruction.

**(Oct-2004)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI H, 1050 H	; Initialize H-L pair with starting address
	LXI D, 1070 H	; Initialize D-E pair with destination address
	MOV C, 0AH	; Get count in register C
UP	MOV A, M	; Get number from memory to acc.
	XCHG	; Exchange the memory pointer (HL ↔ DE)
	MOV M, A	; Transfer the number at destination memory.
	XCHG	; Exchange memory pointer for original position
	INX H	; Increment memory pointer
	INX D	; Increment destination memory pointer
	DCR C	; Decrement count
	JNZ UP	; Is zero ? NO - jump to label UP
	HLT	; Stop processing

- 35) Write a program in assembly language to find the two's compliment of a sixteen bit number stored in memory location C000 and C001. Store the result in memory locations C002 and C003.

**(Oct-2004)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI H, C000H	; Point to LSBs of 16-bit number
	MVI B, 00H	; Move immediate 00H content to B reg.
	MOV A, M	; Get 8-LSBs in Accumulator
	CMA	; Take 1's complement of 8-LSBs of number
	ADI 01 H	; Add 01 for 2's complement
	STA C003 H	; Store the 8-LSBs result
	JNC GO	; Is carry ? No-jump to label GO
	INR B	; If carry then increment B by 1
GO	INX H	; Point to 8-MSBs of 16-bit number
	MOV A, M	; Get number in A
	CMA	; Take 1's complement of 8-MSBs
	ADD B	; If carry from 8-LSBs then add it to MSBs
	STA C004 H	; Store the 8-MSBs result
	HLT	; Stop processing

- 36) Write an assembly language program to check validity of given code at location C020. A code is said to be valid, if count of high (logic 1) in first five MSB's reads two and remaining three bits read low.

If code is valid, HL should read AAAA or else it should read FFFF.

[Oct-2004]

Ans.:

Label	Mnemonics+ Operand	Comments
START	MVI B, 00 H	; Set count = 00H
	MVI C, 05 H	; Set count = 05 H
	LXI H, C020 H	; Initialize H - L pair
	MOV A, M	; Get data into A
AGAIN	RLC	; Rotate contents of A left side
	JNC ZERO	; If bit is low then jump to ZERO
	INR B	; Increment count if bit is high
	DCRC	; Decrement count
	JNZ AGAIN	; Is zero ? No, jump to AGAIN
	MOV A, B	; Move high bit count to A
	CPI 02 H	; Check count is 2 or not
	JNZ END	; If no, then jump to END
	MOV A, M	; If yes, take original data to check
		; last three bits
UP	MVI C, 03 H	; Set count C = 03 H
	RRC	; Rotate contents of Acc right side
	JC END	; If bit is high then invalid data
	DCRC	; If low then decrement count
	JNZ UP	; Repeat until C ≠ 0
	LXI H, AAAA H	; Valid data so set HL with AAAAA
	JMP STOP	; Goto Halt
END	LXI H, FFFF H	; Invalid data, set HL with FFFF
STOP	HLT	; Stop processing

- 37) Write an assembly language program to count the number of times a data D5 H is found in a block of memory having starting address 3000 H. Length of the block is stored in 2FFF H. Store the result in memory location 2000 H.

[March 2005]

Ans.:

Label	Mnemonics+ Operand	Comments
START	MVI B, 00 H	; Set count = 00 H
	LXI H, 2FFF H	; Set HL pointer to 2FFF H
	MOV C, M	; Get count in register C
LOOP	INX H	; Increment H-L pair

Label	Mnemonics+ Operand	Comments
	MOV A, M	; Get number in accumulator
	CPI D5 H	; Check whether the number is D5 or not.
	JNZ NEXT	; If not zero ? Jump to NEXT
	INR B	; If found then increment count
NEXT	DCR C	; Decrement count
	JNZ LOOP	; Repeat Loop if count $\neq$ 0
	MOV A, B	; Store count in A
	STA 2000 H	; Store count in 2000 H
	HLT	; Stop processing

- 38) Write an assembly language program to get a decimal sum of series of numbers whose length is stored in C000 H and series itself starts from C001 H. Store the result in C050 H and C051H. **(March, 2005)**

Ans. :

Label	Mnemonics+ Operand	Comments
START	LXI H, C000H	; Set HL pointer to C000H
	MOV C, M	; Get count in register C
	MVI A, 00H	; Make LSBs of SUM = 00 H
	MOV B, A	; Make MSBs of SUM = 00 H
LOOP	INX H	; Set HL to point the number in series
	ADD M	; Previous no. + Next no.
	JNC AHEAD	; Is carry ? No goto AHEAD
	INR B	; Yes, add carry to MSBs of sum
AHEAD	DAA	; Adjust accumulator to decimal contents
	DCR C	; Decrement count
	JNZ LOOP	; Is count = 0 ? No, jump to LOOP
	STA C050 H	; Store LSBs of the sum to C050H
	MOV A, B	; Get MSBs of sum in accumulator
	STA C051 H	; Store MSBs
	HLT	; Stop processing

- 39) Write a program in assembly language to find the smallest number from a serial of numbers, whose length is stored in C000 H and the series itself begins from C001 H. Store the result in memory location C050 H. **(March 2005)**

Ans.:

Label	Mnemonics+ Operand	Comments
	START LXI H, C000H	: Set HL pointer to C000H
	MOV C, M	: Get count in C
	MVI A, FF H	: Set smallest = FF H
LOOP	INX H	: Point to number in series
	CMP M	: Compare with previous number Is it smaller?
	JC AHEAD	: No, smaller is in accumulator. Goto AHEAD
	MOV A, M	: Yes, get smaller no. in accumulator
	DCR C	: Decrement count
	JNZ LOOP	: Repeat if count <input type="checkbox"/> 0
	STA C050H	: Store the result
	HLT	: Stop processing

- 40) Write an assembly language program to separate two nibbles of an 8-bit number stored in memory location 1500H. Add these two nibbles and store the sum in memory at BABAH location. **(October 2005)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	LDA 1500H	: Load operand in Accumulator
	MOV B, A	: Get the number in B register
	RRC	: With four rotate
	RRC	: instructions make
	RRC	: 4 MSBs bits to
	RRC	: 4 LSBs and vice-versa.
	ANI 0FH	: Mask off 4 MSB's
	MOV C, A	: Move contents of accumulator into C register
	MOV A, B	: Move contents of B register into accumulator
	ANI 0FH	: Mask off 4 MSB's
	ADD C	: Add contents of register C to accumulator
	STA BABAH	: Store result at BABAH
	HLT	: Stop program execution



- 41) Write an assembly language program to convert the given 8-bit number stored in memory location ABCDH into ASCII format and store the ASCII value at location ABCEH and ABCEH.

(October, 2005)

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI H, ABCDH	; Address of given no. in HL
	LXI D, ABCEH	; Destination address
	MOV A, M	; Move given number in accumulator
	RRC	; With four rotate
	RRC	instruction make
	RRC	4 LSBs to 4 MSBs
	RRC	and vice-versa
	CALL CON	; Call conversion subroutine
	STAX D	; Store ASCII value of 4 MSB's
	INX D	; Increment DE pointer
	MOV A, M	; Get original number into accumulator
	CALL CON	; Call conversion subroutine
	STAX D	; Store ASCII value of 4 LSBs
	HLT	; Stop program execution
CON	ANI 0FH	; Mask off 4 MSBS
	CPI 0AH	; Compare accumulator with OA
	JC NEXT	; If carry ? Yes, Jump to NEXT
	ADI 07H	; Add 07H to ACC
NEXT	ADI 30H	; Add 30H to accumulator if digit from 0 to 9
	RET	; Return

- 42) Write an assembly language program to check the validity of each number of the given series. The series is stored in memory location starting from ABCDH to ABDDH. A number is said to be valid if 4 LSB's are greater than 4 MSB's. If the number is valid, then store 11H on that location. Otherwise store 00H on the same location.

(October, 2005)



Ans. :

Label	Mnemonics+ Operand	Comments
START	MVI B, 11H LXI H, ABCDH	; Store count in register B ; Set starting address of series in HL
BACK	MOV A, M RRC RRC RRC RRC	; Get num in accumulator ; With four rotate instruction make 4 LSB's to 4 MSB's and vice-versa
	ANI 0FH	; Mask off 4 MSB's
	MOV D, A	; Move ACC to D
	MOV A, M	; Move original number into accumulator
	ANI 0FH	; Mask off 4 MSB's
	CMP D	; Compare 4 LSB's with 4 MSB's
	JC NEXT	; If carry ? Invalid number Jump to NEXT
	MVI M, 11H	; Otherwise mark this location as valid
	JMP AHEAD	; Jump to AHEAD
NEXT	MVI M, 00H	; Mark this location as invalid
AHEAD	INX H DCR B JNZ BACK HLT	; Increment HL pointer ; Decrement counter ; Jump if no zero to BACK ; Stop program execution B

- 43) Write an assembly language program to perform the addition of 06H data to accumulator if auxiliary carry flag is set. Store this sum in memory at BABAH.

Oct. 05

Ans. :

Label	Mnemonics+ Operand	Comments
START	LXI SP, BBBFH PUSH PSW POP H	; Initialize stack pointer ; Store contents of ACC and Flag register into stack ; Retrieve the current contents of ACC and Flag and store in H and L
	MOV A, L	; Move content of L (flag register) to accumulator
	ANI 10 H	; AND immediate ACC to 10 H
	JZ AHEAD	; if zero ? Jump to AHEAD
	ADI 06 H	; AC flag set so add 06 to ACC.
AHEAD	STA BABAH HLT	; Store the result ; Stop program execution

- 44) A set of eight data bytes are stored in memory starting from ABCDH. Write an assembly language program to add two bytes at a time and store the sum in the same memory location, low order sum replacing the first byte and carry replacing second byte. If any pair does not generate a carry, the memory location of second byte should be cleared. **(October 2005)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	MVI C, 04H	; Store count in register C
	LXI H, ABCDH	; Load HL pair with memory address ABCDH
	MOV A, M	; Move contents of memory to ACC
	INX H	; Increment memory pointer
	ADD M	; Add second byte with first byte
	DCX H	; Decrement memory pointer
	MOV M, A	; Replace the first byte by SUM
	INX H	; Increment memory pointer
	JC NEXT	; Jump if carry to NEXT
	MVI M, 00H	; Otherwise clear memory holding second byte
	IMP AHEAD	; Jump on AHEAD
NEXT	MVI M, 01H	; Store the carry at memory holding second byte
	INX H	; Increment memory pointer
AHEAD	DCR C	; Decrement counter
	JNZ LOOP	; Jump if not zero to LOOP
	HLT	; Stop program execution

- 45) Write a subroutine labelled 'FIND' to search the largest element from a given unsigned series stored in memory location from ABBAH to ABCDH. Store the largest element at ABCEH and its address in HL register pair. **(October 2005)**

Ans.:

Label	Mnemonics+ Operand	Comments
FIND	LXI SP, 2000 H	; Initialize stack
	LXI H, ABBAH	; Load HL pair with ABBA H
	MVI A, 00H	; Clear ACC
	MVI C, 14H	; Store count in register C
AGAIN	CMP M	; Compare ACC with memory
	JNC NEXT	; Is No carry ? Jump to NEXT
	MOV A, M	; Otherwise take larger number from memory to ACC
	PUSH H	; Store the address of larger number in stack
	INX H	; Increment memory pointer
	DCR C	; Decrement counter
	JNZ AGAIN	; Jump if no zero to AGAIN
	STA ABCEH	; Store larger number in memory
	POP H	; Retrieve and hold address of larger number in HL
NEXT	RET	; Return

- 46) Write an assembly language program to add all even numbers stored in a memory block of 10 locations starting from 2000H, store the two byte sum at memory location starting from 3000H. **(March 2006)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	MVI C, 0AH	; Initialize counter
	LXI H, 2000H	; Initialize pointer
	MVI E, 00H	; Sumlow = 0
	MOV D, E	; Sumhigh = 0
	MOV A, M	; Get the number in accumulator
	ANI 01 H	; Mask bit 1 to bit 7
	JNZ SKIP	; Don't add if number if ODD
	MOV A, E	; Get the lower byte of SUM
	ADD M	; SUM = SUM + data
	MOV E, A	; Store result in E register
	JNC SKIP	; If no carry ? Go to SKIP
	INR D	; Add carry to MSB of SUM
	INX H	; Increment pointer
	DCR C	; Decrement counter
	JNZ BACK	; Check if counter ≠ 0 repeat
	MOV A, E	; Get LSBs in ACC
	STA 3000 H	; Store lower byte
	MOV A, D	; Get MSBs in ACC.
	STA 3001 H	; Store higher byte
	HLT	; Terminate program execution

- 47) Write a program to set the sign and zero flag bits of the flag register to '1' and reset to '0' the remaining flag bits. The content of accumulator should be AAH. Also the content of BC, DE and HL register pair should be same as that of PSW. **(March 2006)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI SP, 2000 H	; Set stack pointer
	PUSH PSW	; Save flags on stack
	POP H	; Retrieves flags in L register
	MVI A, AAH	; Get contents in ACC
	ANI 80 H	; Set sign and zero flag and reset remaining flag bits.
	MOV L, A	; Transfer ACC to L register



Label	Mnemonics+ Operand	Comments
	PUSH H POP PSW MOV B, H MOV C, L MOV D, H MOV E, L HLT	; Save flags on stack ; Retrieves flag in flag register. ; Transfer PSW contents into B and D register pair  Stop processing

- 48) Write an assembly language program to fill up the memory block of 20 memory locations starting from 2000H, with data bytes 00H and FFH at every alternate memory locations.

Ans. :

Label	Mnemonics+ Operand	Comments
START	MVI C, 13 H MVI D, 00H MVI E, FFH LXI H, 2000H MOV M, D DCR C INX H MOV M, E INX H DCR C JNZ UP HLT	; Initialize counter ; Move 00 to D register ; Move FF to E register ; Initialize pointer ; Fill 00 in memory location ; Decrement count ; Increment pointer ; Fill FF to next memory location ; Increment pointer ; Decrement count ; Check if counter $\neq$ 0 repeat ; Stop processing
UP		

OR

Label	Mnemonics+ Operand	Comments
START	MVI C, 13 H LXI H, 2000H MOV A, L ANI 01 H JNZ ODD MVI M, 00 H JMP LOOP MVI M, FF INX H DCR C JNZ UP HLT	; Initialize counter ; Initialize pointer ; Get contents of register L in ACC ; mask bit 1 to bit 7 ; Jump if odd memory location num. ; Fill 00 H in memory ; Goto Loop ; Fill FFH in memory. ; Increment pointer ; Decrement counter ; Check if counter $\neq$ 0 repeat ; Stop processing
UP		
ODD LOOP		

- 49) A three byte number is stored in a memory with starting address 2000H. Write a program to check whether it is palindrome or not. If it is palindrome, then store 00H in register B else store FFH.

(March, 2006)

Ans. :

Label	Mnemonics+ Operand	Comments
START	LXI H, 2000 H	; Initialize pointer
	LXI D, 2002H	; Initialize DE pair
	MOV A, M	; Get first byte of number
	RRC	; With 4 rotate
	RRC	instruction exchange
	RRC	the two nibbles of number
	RRC	; Exchange contents of HL and DE
	XCHG	; Check if first and last byte of number is same or not
	CMP M	; If not same, go to ESCAPE
	JNZ ESCAPE	; Reexchange contents of HL and DE
	XCHG	; Point to next byte
	INX H	; Get middle byte of number in ACC
	MOV A, M	; Mask off 4 MSBs
	ANI 0F	; Store into C register
	MOV C, A	; Get original number
	MOV A, M	; With 4 rotate instruction exchange
	RRC	two nibbles of number
	RRC	
	RRC	
	RRC	
	ANI 0F	; Mask off 4 MSBs
	CMP C	; Check two nibbles of number
	JNZ ESCAPE	; If not same goto ESCAPE
	MVI B, 00H	; Set 00 into B register if number is palindrome.
	JMP STEP	; Jump unconditionally to STEP.
ESCAPE	MVI B, FFH	; Set FFH into B register if number is not palindrome
STEP	HLT	; Stop processing

- 50) A BCD number is stored at memory location 2000H. Write an assembly language program to convert it into hexadecimal number and store it is the next memory location.

(March, 2006)



Ans. :

Label	Mnemonics+ Operand	Comments
START	LXI H, 2000 H	; Initialize pointer
	MOV A, M	; Get BCD number
	ANI 0FH	; Mask most significant four bits
	MOV C, A	; Save unpacked BCD1 in C
	MOV A, M	; Get BCD again
	ANI 0FH	; Mask least significant four bits
	RRC	; With 4 rotate instruction
	RRC	convert most
	RRC	significant four bits into
	RRC	unpackaged BCD2
	MOV D, A	; Save BCD2 in D
	XRA A	; Clear accumulator
	MVI E, 0AH	; Set E as multiplier of 10
SUM	ADDE	; Add 10 until [D] = 0
	DCR D	; Reduce BCD2 by one
	JNZ SUM	; Is multiplication complete ? If not go back and add again
	ADD C	; Add BCD1
	INX H	; Increment pointer
	MOV M, A	; Store HEX number
	HLT	; Stop processing

Hint : Converting 2-digit BCD number into its equivalent Hex number following steps are requires:

- (1) Separate an 8-bit packed BCD number into two 4-bit unpacked BCD digits. BCD 1 and BCD 2.
- (2) Convert each digit into its binary equivalent according to its position.
- (3) Add both numbers to obtain the Hex equivalent of the BCD number.

For e.g. Convert (72)<sub>BCD</sub> into Hex equivalent

$$72_{10} = 0111\ 0010_{BCD}$$

Step 1 : 0111 0010 → 0000 0010 Unpacked BCD 1

→ 0000 0111 Unpacked BCD 2

Step 2 : Multiply BCD 2 by 10 (7 × 10)

Step 3 : Add BCD 1 to answer in step 2.

- 51) Write an assembly language program to add all odd numbers stored in memory block of 10 locations starting from 2000 H. Store the two byte sum at memory locations starting from 3000H.

**(October 2006)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI D, 0000H MVI C, 0AH LXI H, 2000H	; Initialize sum ; Initialize counter ; Initialize pointer
NEXT	MOV A, M RRC JNC SKIP MOV A, E ADD M JNC SKIP1 INR D	; Get the number in accumulator ; Check LSB ; Don't add if number is even ; Get the lower byte of sum ; SUM = SUM + data ; If no carry ? Go to SKIP1 ; Add carry to MSB of sum
SKIP1	MOV E, A	; Store result in E register
SKIP	INX H DCR C JNZ NEXT XCHG	; Increment pointer ; Decrement counter ; Check if counter $\neq$ 0 repeat ; Get sum in HL ; Store sum
	SHLD 3000H HLT	; Terminate program execution

- 52) Write an assembly language program to find the sum of first 10 numbers of the series  $2^0, 2^1, 2^2, 2^3, \dots$  Store the two byte result at memory locations starting from address 2000 H.

**(Oct. 2006)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	MVI C, 0AH LXI H, 0000H LXI D, 0001H	; Initialize counter ; Initialize sum ; Initialize first term i.e. $2^0 = 1$
NEXT	DADD D XCHG DADD H XCHG DCR C JNZ NEXT SHLD 2000H HLT	; Add sum term ; Exchange data ; Calculate next term ; Exchange data ; Decrement count ; If counter $\neq$ 0, repeat ; Store result ; Stop

- 53) Write an assembly language program to find the occurrence of numbers divisible by 4 in a memory block of 10 locations starting from 2000 H. Store the count of occurrence at the end of block.

**(October 2006)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	MVI B, 00H MVI C, 0AH LXI H, 2000H	; Initialize count ; Initialize counter ; Initialize pointer
NEXT	MOV A, M RRC JC SKIP RRC JC SKIP INR B INX H DCR C JNZ NEXT MOV M, B HLT	; Get number in accumulator ; Check LSB ; Don't count if no. is not divisible by 4 ; Check second LSB ; Don't count if no. is not divisible by 4 ; Increment count if no. is divisible by 4 ; Increment pointer ; Decrement counter ; Repeat if counter $\neq$ 0 ; Store result at end of block ; Terminate process

54) Write an assembly language program to generate the first ten numbers of Fibonacci series and store them in a memory block starting from 2000 H. **(October. 2006)**

Ans.:

Label	Mnemonics+ Operand	Comments
START	LXI H, 2000H MVI D, 08H LXI B, 0100H MOV M, C INX H MOV M, B INX H MOV A, C ADD B MOV M, A MOV C, B MOV B, A DCR D JNZ NEXT HLT	; Initialize pointer ; Initialize counter ; Initialize B = 01 and C = 00 ; Copy first term at memory ; Increment HL pair ; Copy second term at memory ; Increment HL pair ; Get previous term in accumulator ; Generate next term by adding ; previous two terms ; Copy result at memory ; Copy B to C ; Copy new term to B register ; Decrement counter ; Repeat if counter $\neq$ 0 ; Stop processing

55) An ASCII code for a hexa-decimal digit is stored at memory location 2000 H. Write an assembly language program to convert it into hexa-decimal number and store it at 3000 H. **(October. 2006; July 18)**

Ans. :

Label	Mnemonics+ operand	Comments
	LDA 2000	; Get ASCII code in accumulator
	SUI 40H	; Subtract 40H from acc
	JM NUM	; If code < 40, goto NUM
	ADI 09H	; If code > 40, add 09 to acc
	JMP OUT	; Jump at OUT
	ADI 0AH	; Add 0A if code < 40
	STA 3000H	; Store result
	HLT	; Stop processing

- 56) Write an assembly language program to count the number of 1's and 0's in a 8 bit binary number stored at memory location 2000 H. Store the counts for 0's and 1's in a memory location 2001 H and 2002 H respectively.

**October, 2008**

Ans. :

Label	Mnemonics+ operand	Comments
START	LXI H, 0000H	; Clear H-L pair
	LDA 2000H	; Get number in accumulator
	MOV C, 08	; Initialize counter
NEXT	RRC	; Rotate acc to check a bit
	JNC SKIP	; If bit is zero, goto SKIP
	INRL	; Increment count since bit is 1
	DCR C	; Decrement counter
	JNZ NEXT	; Repeat if counter $\neq 0$
	MVI A, 08	; Initialize acc
	SUBL	; Subtract count of 1 bit
	MOV H, A	; Take count of num of 0 bit
	SHLD 2001H	; Store result
	HLT	; Stop processing

- 57) Write an assembly language program to shift 16-bit number by three bit left, store in memory location starting from BABAH with LSB. Store the result startin from BADAH.

**March 2010**



Ans:

Label	Mnemonics+ operand	Comments
START	LHLD BABAH DAD H JNC AHEAD INR L DAD H JNC NEXT INR L	; Load 16-bit number in HL register pair ; Double add ; Jump, if no carry AHEAD ; if carry, increment lower byte by one ; Double add ; Jump, if no carry to NEXT ; If carry, increment lower byte by one ; Double add
AHEAD	DAD H JNC NEXT INR L	; Jump, if no carry to 'FORWARD', ; If carry, increment lower byte by one ; Store result
NEXT	DAD H JNC FORWARD INRL SHLD BADAH	; Jump, if no carry to 'FORWARD', ; If carry, increment lower byte by one ; Store result
FORWARD	SHLD BADAH HLT	; Stop program execution

56) Write an assembly language program to sort 25 numbers in-ascending order stored in memory location from AB01H and onward. Store the sorted data in memory from BC01H and onward. **(March, 2007)**

Ans:

Label	Mnemonics+ operand	Comments
START	LXI H, AB01H MVI C, 19H LXI D, BC01H BACK MOV A, M STAX D INX H INX D DCR C JNZ BACK	; Load HL pair with AB01H. ; Initialize Count ; load DE pair with BC01H ; Move memory to Acc. ; Store Acc. indirect ; Increment HL pair pointer ; Increment DE pair pointer ; Decrement Count ; Jump, if no zero to BACK.
UP2	MVI B, 19H LXI H, BC01H MVI C, 19H MOV A, M INX H CMP M JC DOWN MOV D, M MOV M, A DCX H MOV M, D INX H DCR C JNZ UP1	; Load HL pair with BC01 H ; Move imm. 19H in reg. C ; Move memory to Acc. ; Increment HL pair pointer ; Compare Acc. with memory ; Jump on carry to DOWN ; Move memory to reg. D ; Move Acc., to memory ; Decrement HL pair pointer ; Move reg. D to memory ; Increment HL pair pointer ; Decrement Count ; Jump, if no zero to UP1 ; Decrement reg. B by one ; Jump, if no zero to UP 2 ; Stop program execution
UP1	MOV A, M INX H CMP M JC DOWN MOV D, M MOV M, A DCX H MOV M, D INX H DCR C JNZ UP1 DCR B JNZ UP2 HLT	
DOWN	DCR C JNZ UP1 DCR B JNZ UP2 HLT	

- 59) Write an assembly language program to fill the memory location starting from ABOOH and onward with decimal number from 0 to 99. **(March, 2007)**

Ans :

Label	Mnemonics+ operand	Comments
START	LXI H, AB00H MVI B,64H SUB A MOV M, A ADI 01H DAA INX H DCR B JNZ BACK HLT	; Load HL pair with ABOOH. ; Store memory Block count in B ; Clear accumulator I ; Move accumulator to memory ; Add 01 to accumulator ; Decimal adjust accumulator ; Increment HL memory pointer ; Decrement counter by one ; Jump, if no zero to BACK ; Stop process

- 60) Write an assembly language program to find greatest and smallest from a given series stored in memory location from BABAH to BADAH Store the smallest number at BADBH and greatest number at BADCH. **(March, 2007)**

Ans :

Label	Mnemonics+ operand	Comments
START	LXI H, BABA MVI B, 21H DCR B MOV A, M MOV D, A MOVE, A INX H CMP M JC LARGE MOV D, A MOV A, M CMP E JNC SMALL MOV E, A MOV A, D JMP XI MOV A, M DCR B JNZ BACK STA BADCH MOV A, E STA BADB HLT	; Load HL pair immediate with BABA ; Move counter in B ; Decrement counter ; Move memory to Acc ; Move Acc to D ; Move Acc to E ; Increment HL pair pointer ; Compare memory with Acc ; Jump, if carry to LARGE ; Otherwise move A to D ; Move memory to Acc. ; Compare E with Acc ; Jump, if no carry to SMALL ; Move Acc to E ; Move D to Acc ; Jump to XI ; Move memory to Acc. ; Decrement B ; Jump, if no zero to BACK ; Store Acc at BADCH ; Move E to A ; Store Acc at BADB ; Stop execution

- 61) Write an assembly language program to add two 8-bit numbers stored in memory location ABCDH and ACEH. Store the sum in memory at ABDDH and the flag status at location ABDCH. **(March, 2007)**

Ans:

Label	Mnemonics+ operand	Comments
START	LXI SP 2000H LHLD ABCDH MOV A, H ADD L PUSH PSW POP H ABDCH HLT	; Initialize stack pointer ; Load numbers in H and L ; Move one operand to Ace ; Add second operand ; Push PSW to stack ; Retrieve result in HL ; Store result ; Stop process

- 62) Squares of decimal numbers from 0 to 9 are stored in memory from 1500H to 1509H respectively. Write an assembly language program to find the square of a given decimal number by look up table method given in the above range and is stored at 14F2H. Store the square of given number in memory at 14F3H. **(March, 2007)**

Ans:

Label	Mnemonics+ operand	Comments
START	LDA 14F2H MOV L, A, MVI H, 15 H MOV A, M STA 14F3H HLT	; Load Acc direct from memory ; Move A to L ; Move immediate 15 H to H ; Move memory to Acc. ; Store result ; Stop process

- 63) Write an assembly language program to count number of odd data bytes in the block of memory starting from 1300H to 13FFH and output on port 11H. **(October, 2007)**

Ans:

Label	Mnemonics+ operand	Comments
START	STC CMC LXI H, 1300 H MVI B, FFH MVI C, 00H MOV A, M RAR JNC Skip INR C	; Set carry ; Clear carry ; 1300 H Set HL pointer to 1300 H ; FFH Set location counter to FFH ; Set odd bytes counter to 00H ; get data from memory ; Rotate acc. Right through carry ; Skip check odd or even ; if odd byte counter increments



Label	Mnemonics+ operand	Comments
Skip	INX H DCR B JNZ Again MOV A, M RAR JNC Next INRC	; increments memory pointer ; Decrement B reg content by 1 ; checks if 255 bytes are checked ; Loads 256th byte if 255 bytes are checked ; Rotate acc. Right through carry ; Checks odd or even ; if odd, increment odd ; byte counter & copies it to ; accumulator which is then output to output port 11H. ; Stop processing
Next	MOV A, C OUT 11H	
END	HLT	

64) Write a program segment using appropriate 'Rotate' instruction to divide the number in BC register pair by 2. The quotient should remain in BC register pair. (October, 2007)

Ans :

Label	Mnemonics+ operand	Comments
START	STC CMC MOV A, B RRC MOV B, A RRC MOV B	; Set carry ; complement the carry } set carry to 0 ; Carry MSBs to acc ; Carry to MSB & LSB of higher order. ; Carry LSBs to acc ; Rotate acc to right by 1 ; move acc content to B reg. stop processing
END	HLT	

65) Write an assembly language program to count how many times 05H comes in memory block starting at 4000H to 4004H. Store the result at 4070H. (October, 2007)

Ans :

Label	Mnemonics+ operand	Comments
	LXI H, 4000 H MVI C, 00 H MVI D, 05 H MVI A, 05 H MOV B, M SUB B JNZ loop INRC	; Load the HL pair at 4000 ; clear the C Reg. for counter ; store the total No. 05 in D register. ; Store no. 05 in A reg. ; Transfer the contents of memory to B register. ; Subtract the value of B reg. from A. ; If A # 0 then go to Loop ; Otherwise increment the counter by 1.
Start		



