Page No.
Date

Assignment No. 4

Que-1 Explain Instruction format of 8085.

Ans -

Intel 8085 is an 8 bit microprocessor. It can handle 8 bits of data at a time. One byte consists of 8 bits. A memory location for Intel 8085 microprocessor is designed to accommodate 8 bit data. If a data 16 bit is to be stored, it is stored in two consecutive memory locations.

Due to different ways of specifying data for the instructions, the machine codes of all the instructions are not of the same length. Depending upon the length of machine code of the instructions, instructions can be classified in three types :-

i) One byte Instructions
ii) Two byte Instructions
iii) Three byte Instructions

i) **One byte Instructions -**

These include the operation code and the operand in the same byte. The length of the machine code for these instructions is of one byte. Eg. MOV C, A ADD B etc

ii) Two Byte Instructions -

In this, the first byte specifies the operation code and the second byte specifies the operand which is an 8-bit data or an address.

Opcode	Data / Address

iii) Three Byte Instructions -

In these, the first byte specify the operation code and the second and third byte specify the 16 bit data or 16 bit address.

Opcode	Data / Address	Data / Address

Que-2 Explain the Addressing modes of 8085.

Ans - Data which is to be operated on is stored in the memory location or successive memory locations. Each instruction requires certain data on which it has to operate. There are various techniques to specify the address of the data. These techiques are called Addressing modes.

Intel 8085 uses the following Addressing modes-

i) Immediate Addressing Mode
ii) Register Addressing Mode
iii) Direct Addressing Mode

iv) Indirect Addressing Mode
v) Implied Addressing Mode

Que-3 Explain the Data transfer group, Arithmetic Group, Logic Group, Stack I/O and Machine Control.

- Group instructions of 8085 as under

Sr. No.	Syntax	Example	Length	Addressing Mode	Description

Ans -

Data Transfer Group :

This group of instructions transfer (copies) data from a location called a source to another location called a destination, without modifying the contents of a source. The term data transfer is used for coping data. The contents of the destination are modified where as the contents of a source are not changed and retained without any modifications. E.g. MOV, MVI, etc.

Arithmetic Group :

Intel 8085 microprocessor can perform various arithmetic operations like addition, subtraction, increment, decrement etc. In the addition and subtraction operations, accumulator is one of the operand.

Page No.
Date

Logic Group :

The 8085 microprocessor can perform logical operations like AND, OR, EX-OR, complement etc. All logical operations are performed in relation to the contents of the accumulator and the result of the operation is placed in the accumulator.

Stack I/O :

Stack Input/Output operations can be done by Push/Pop instructions respectively.

PUSH = Push two bytes of data onto the Stack

POP = Pop two bytes of data off the Stack

Machine Control :

It has different instructions which helps in following controlling machine -

HLT = Halt and Enter into wait state

NOP = No operation

etc.

Sr. No.	Syntax	Example	Length	Addressing Mode	Description
1.	ADD R	ADD B Reg. contents before instr. A 47 B 51 Reg. contents after instr. A 98 B 51	1 Byte	Register Addressing Mode	The contents of register R are added to the accumulator and the result after addition is stored in the accumulator.

2.	ADD M	ADD M Reg. contents before instr. A [76] B [20] [50] L Memory content 2050 [A2] Reg. content after instr. A [18] H [20] [50] L	1 Bytes	Register Indirect Addressing Mode	Add memory contents to accumulator.
3.	ADC R	ADC E Reg. contents before instr. A [23H] [4AH] E Reg. contents after instr. A [8EH] [4AH] E	1 Byte	Register Addressing Mode	Add register to accumulator with carry.
4.	ADC H	ADC M contents before A [28H] H [25H] [50H] L Contents after A [A6H] H [25H] [50H] L	1 Byte	Register Indirect Addressing mode	Add the contents of memory to accumulator with carry.
5.	INR M	INR M Contents before H [20] [75] L Content of memory	1 Byte	Register Indirect Addressing Mode	Increment memory contents by 1.

		2075 [7F] Contents after 2075 [80] H [20] [75] L			
6.	DCR M	DCR M Reg. and Memory contents before H [20] [85] L 2085 [A0] After instruction H [20] [85] L 2085 [9F]	1 Byte	Register Indirect Addressing mode	Decrement memory contents by 1.
7.	INX Register Pair	INX H Contents of reg. before H [56] [37] L Contents of reg. after H [56] [38] L	1 Byte	Register Indirect Addressing Mode	Increment register pair by 1.
8.	DCX Register Pair	DCX D Reg: content before inst. D [20] [00] E Reg. content after inst. D [1F] [FF] E	1 Byte	Register Increme Indirect Addressing Mode	Decrement contents of register pair by 1.

9.	MOV rd, rs	MOV B, C Register contents before instruction B 72 9F C Register contents after instruction B 9F 9F C	1 Byte	Register Addressing Mode	Move data from (rs) source register to (rd) destination register.
10.	MVI R, Data / MVI M, data	MVI B, 92H data 92H is copied in the register B.	2 Byte	Immediate Addressing Mode	Move immediate 8-bit data.

Que-4 Write the following assembly language program with neat flowchart for the following

i) Addition of two 8 bit numbers result 8 bit.
ii) Addition of two 8 bit numbers result 16 bit.
iii) Subtraction of two 8 bit numbers.
iv) Multiplication of two 8 bit numbers.
v) Division of two 8 bit numbers.
vi) Arranging 10 numbers in ascending / descending order.

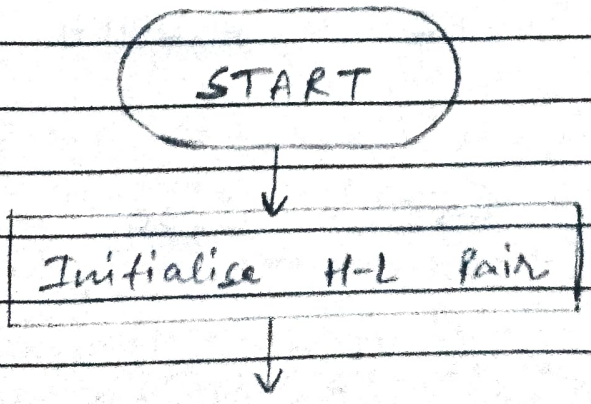
Ans :-

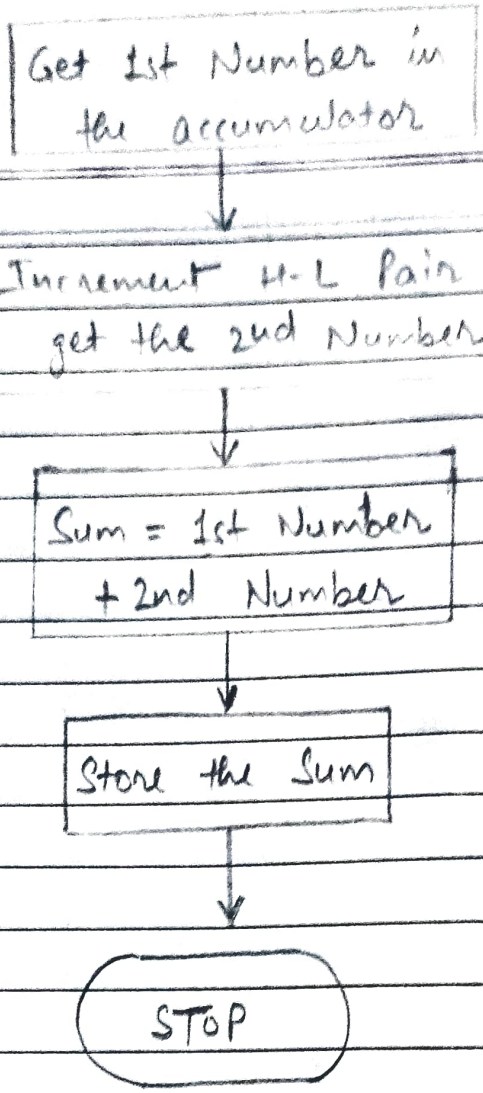
i) Addition of two 8 bit numbers result 8 bit, Add 48 H and 56 H

Program :

Memory Address	Machine Code	Mnemonics	Operands	Comments
2000	21,01,24	LXI	H,2401H	Place address of the 1st number in H-L register pair
2003	7E	MOV	A,M	Move the contents of memory addressed by H-L pair to the accumulator.
2004	23	INX	H	Increased the contents of H-L pair by 1 i.e. address is 2402 in the H-L pair.
2005	86	ADD	M	Add the 2nd number in the first number, and the result is in the accumulator.
2006	32,03,24	STA	2403H	Store the result in memory location 2403.
2009	76	HLT		Stop

Flowchart :





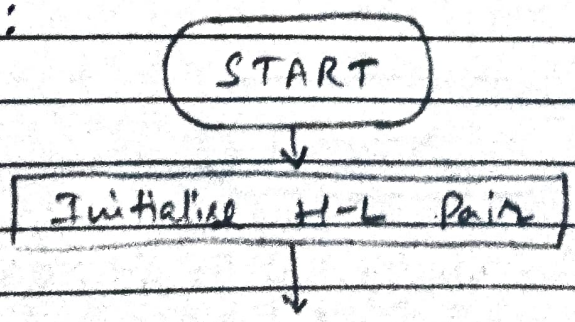
ii) Addition of two 8-bit Numbers, result 16 bit
Add 97H and 9AH

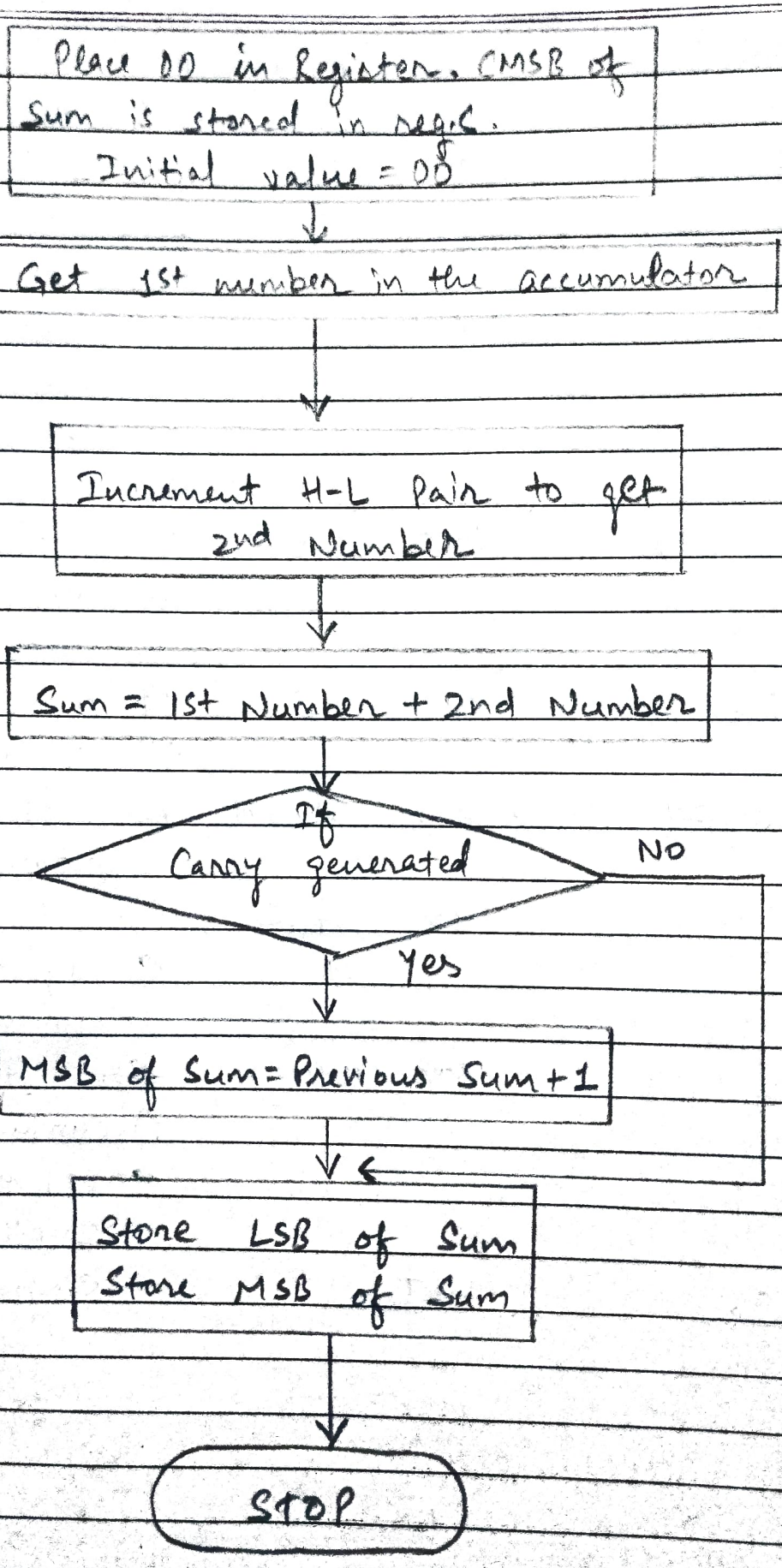
Program:

Memory Address	Machine Code	Label	Mnemonics	Operands	Comments
2000	21,01,24		LXI	H,2401H	Place address of the 1st number in the H-L pair.
2003	0E,00		MVI	C,00	Place 00 in register C. MSB of the sum is to be stored in register C

2005	7E	MOV	A,M	Move the 1st number in the accumulator.
2006	23	INX	H	Add 1 to the H-L register pair, H-L register pair now contain 2402.
2007	86	ADD	M	Add the contents of 2nd number to accumulator and sum is placed in the accumulator.
2008	DC,0C,20	JNC	AHEAD	Jump to label AHEAD, if there in no carry generated.
200B	0B	INR	C	If carry is generated the increment C by 1
200C	32,03,24	AHEAD STA	2403H	Place the sum in memory location 2403.
200F	79	MOV	A,C	Move the contents of C register (carry) to accumulator
2010	32,04,24	STA	2404 H	Store the contents of Accumulator (MSB of SUM) in memory 2404 H
2013	76	HLT		Stop.

Flowchart:




Place 00 in Register. CMSB of Sum is stored in reg.C.
Initial value = 00
Get 1st number in the accumulator
Increment H-L pair to get 2nd Number
Sum = 1st Number + 2nd Number
If Carry generated
No
Yes
MSB of Sum = Previous Sum + 1
Store LSB of Sum
Store MSB of Sum
STOP

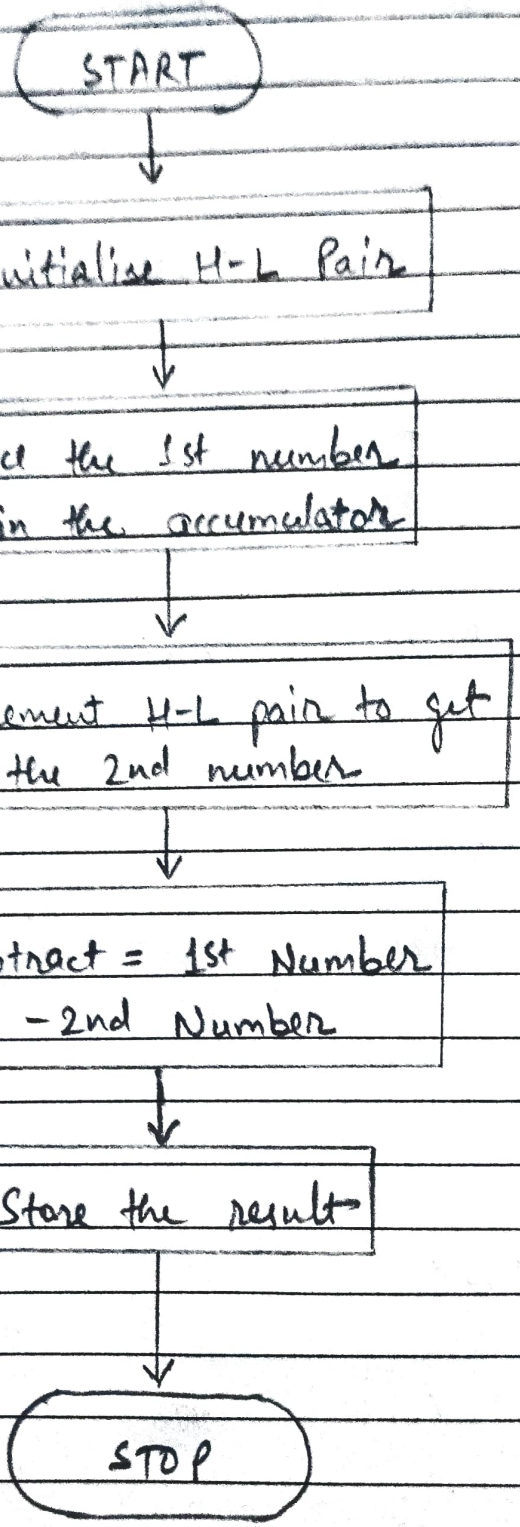
iii) Subtraction of two 8 bit numbers.
Subtract 33H from 48H

Program:

Memory Address	Machine Code	Mnemonics	Operands	Comments
2000	21,01,24	LXI	H, 2401	Load address of first number to the register pair H-L.
2003	7E	MOV	A, M	Move the first number in accumulator
2004	23	INX	H	Add 1 to the contents of H-L register pair.
2005	96	SUB	M	Subtract 2nd number from the first number and the result is placed in the accumulator
2006	23	INX	H	Get memory address 2403 in the H-L pair
2007	77	MOV	M, A	Move the contents of accumulator to the memory location whose address is in the H-L register pair (memory address 2403)
2008	76	HLT		Stop.

Page No.
Date

Flowchart :



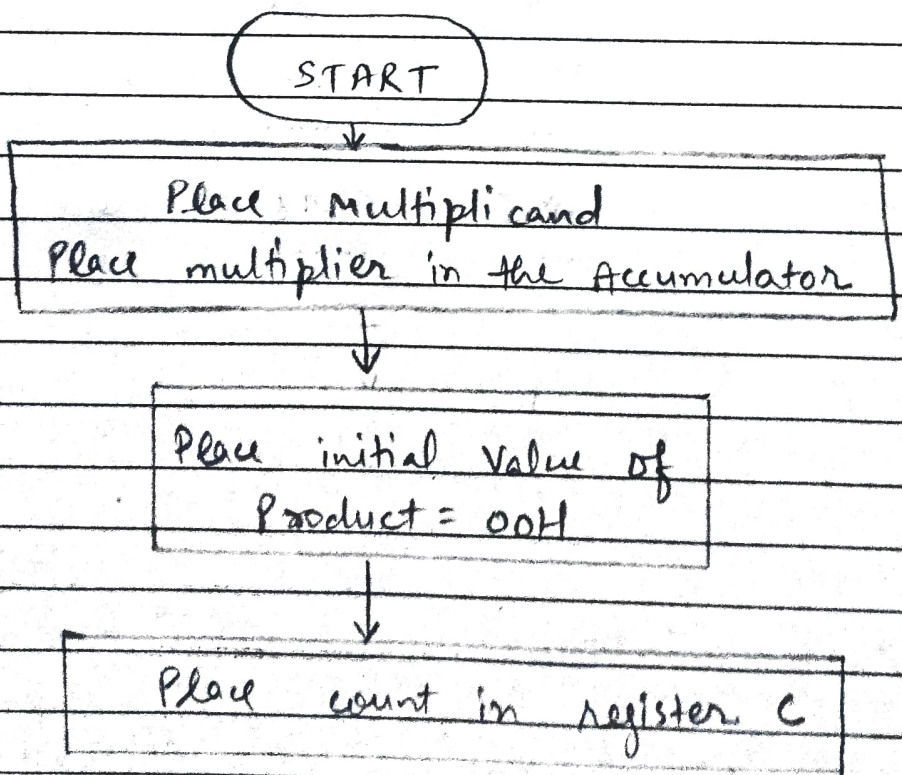
iv) Multiplication of two 8bit Numbers
Multiply A by 8 (Product = 16 bits)
80H by 55H

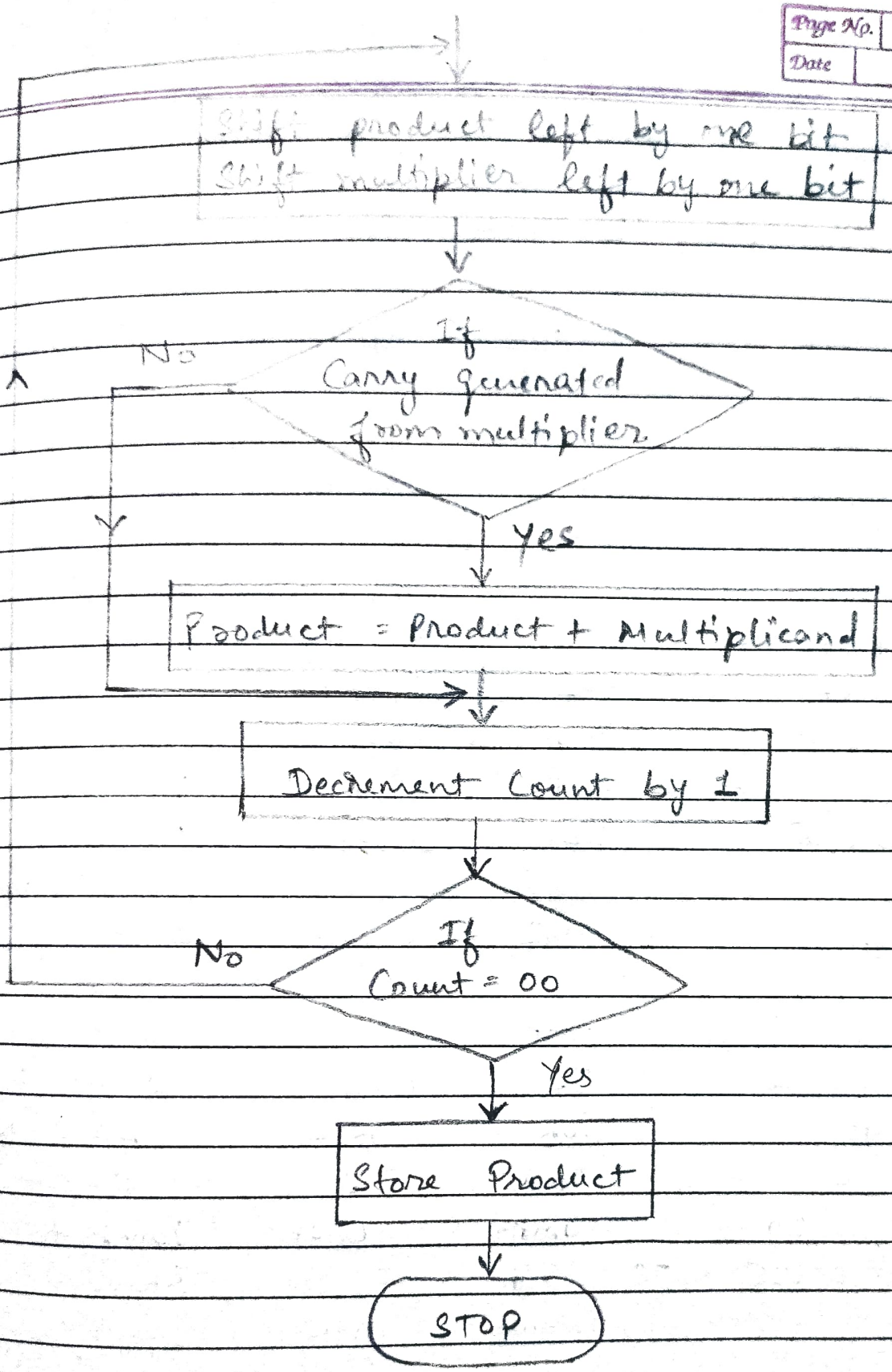
Program:

Memory Address	Machine Code	Label	Mnemonics	Operands	Comments
2000	2A,01,24		LHLD	2401H	Place multiplicand in H-L register pair
2003	EB		XCHG		Place the multiplicand in the register pair D-E.
2004	3A,03,24		LDA	2403H	Get the multiplier in the accumulator.
2007	21,00,00		LXI	H,0000	Place the initial value of the product = 00 in the register pair H-L
200A	0E,08		MVI	C,08	Place value of count = 08 in register C. Since multiplier is of 8 bits and count is equal to the bits of the multiplier.
200C	29	Loop	DAD	H	Add the contents of H-L pair to itself (It will shift the partial product left by 1 bit)

200D	17		RAL	Rotate contents of accumulator which is multiplier, left by 1 bit
200E	D2,12,20		JNC	AHEAD Jump on No carry. If multiplier bit = 1, No, go to label AHEAD.
2011	19		DAD	D If multiplier bit = 1 then Product = Product + multiplicand
2012	0D	AHEAD	DCR	C Decrement Count
2013	C2,0C,20		JNZ	Loop Jump to label loop till C=0
2016	24,04,24		SHLD	2404H Store the result in memory location 2404 H & 2405 H
2019	76		HLT	Stop

A flowchart:




Shift product left by one bit
Shift multiplier left by one bit
If
Carry generated from multiplier
No
Yes
Product = Product + Multiplicand
Decrement Count by 1
If
Count = 00
No
Yes
Store Product
STOP

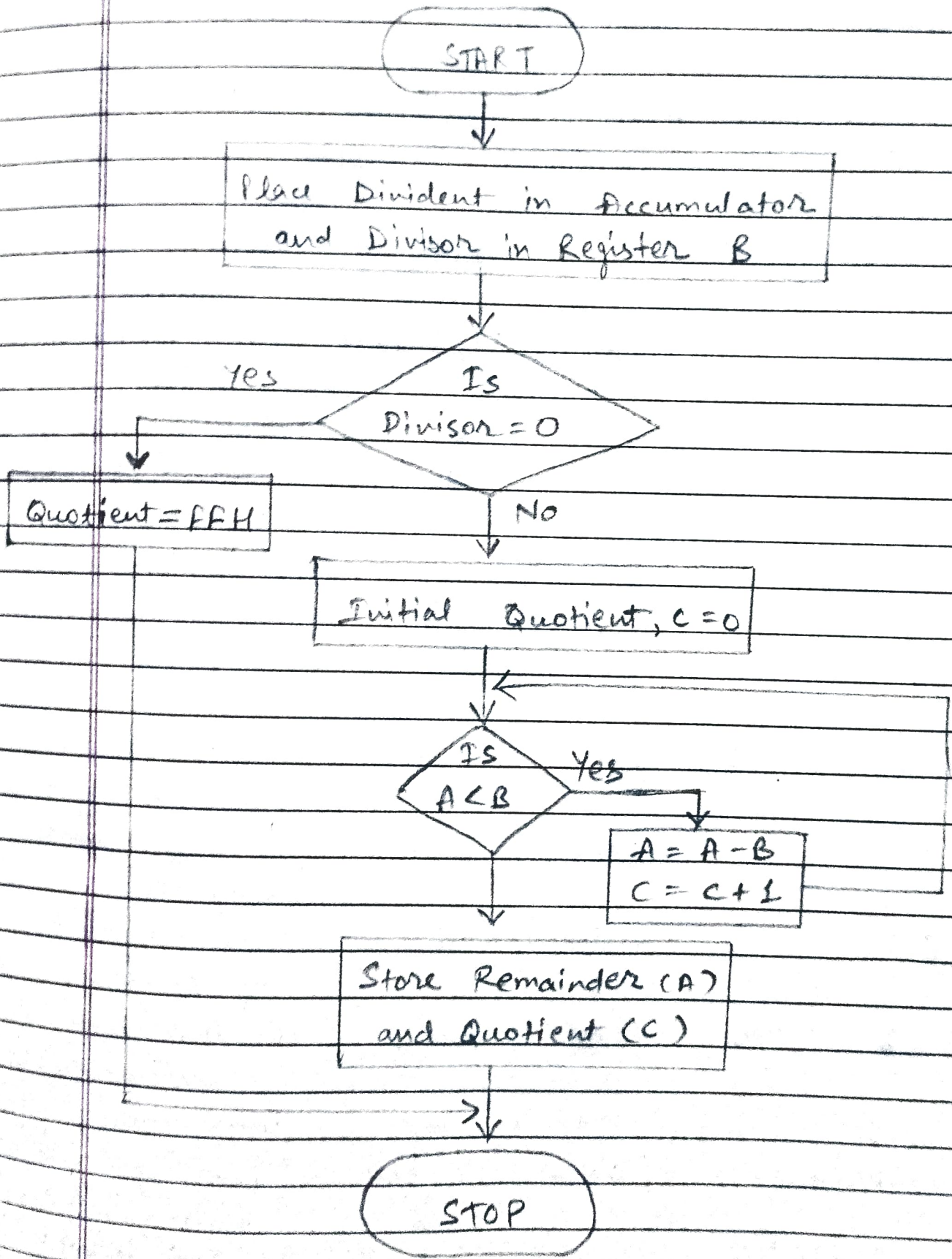
v) Division of two 8 bit Numbers
(8 bit no. by 8 bit no.)
Divide F3H by 14H

Program:

Memory Address	Machine Code	Label	Mnemonics	Operand	Comments
2000	3A,01,25		LDA	2501H	Store Divisor in A
2003	B7		ORA	A	Check whether Divisor = 0
2004	C2,0C,20		JNZ	AHEAD	If not, Jump AHEAD.
2007	0E,FF		MVI	C,FFH	If yes, Quotient = FFH
2009	C3,1B,20		JMP	OVER	Jump to over.
200C	47	AHEAD	MOV	B,A	Copy divisor in B
200D	3A,00,25		LDA	2500H	Store Dividend in Accumulator
2010	0E,00		MVI	C,00H	Initialize Quotient = 00H
2012	BB	LOOP	CMP	B	Check A<B
2013	DA,1B,20		JC	OVER	If yes, jump to over
2016	90		SUB	B	A = A-B
2017	0C		INR	C	C = C-1
201B	C3,12,20		JMP	LOOP	Jump to loop.
201B	32,02,25	OVER	STA	2502H	Store remainder
201E	79		MOV	A,C	Copy Quotient to Accumulator
201F	32,03,25		STA	2503H	Store Quotient
2022	76		HLT		

Page No.
Date

Flowchart :



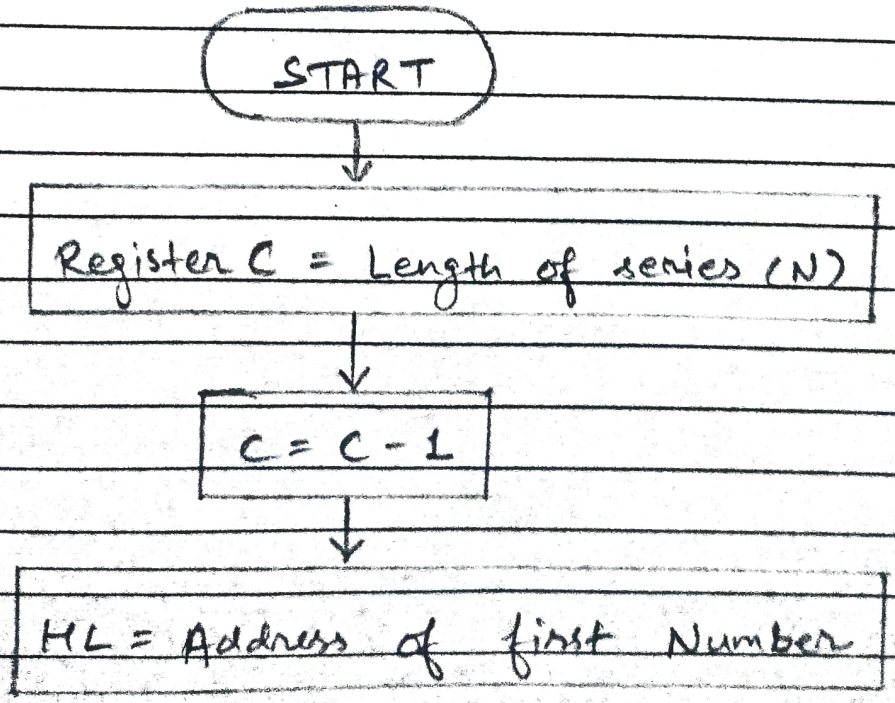
vi) Arranging 10 numbers in descending order

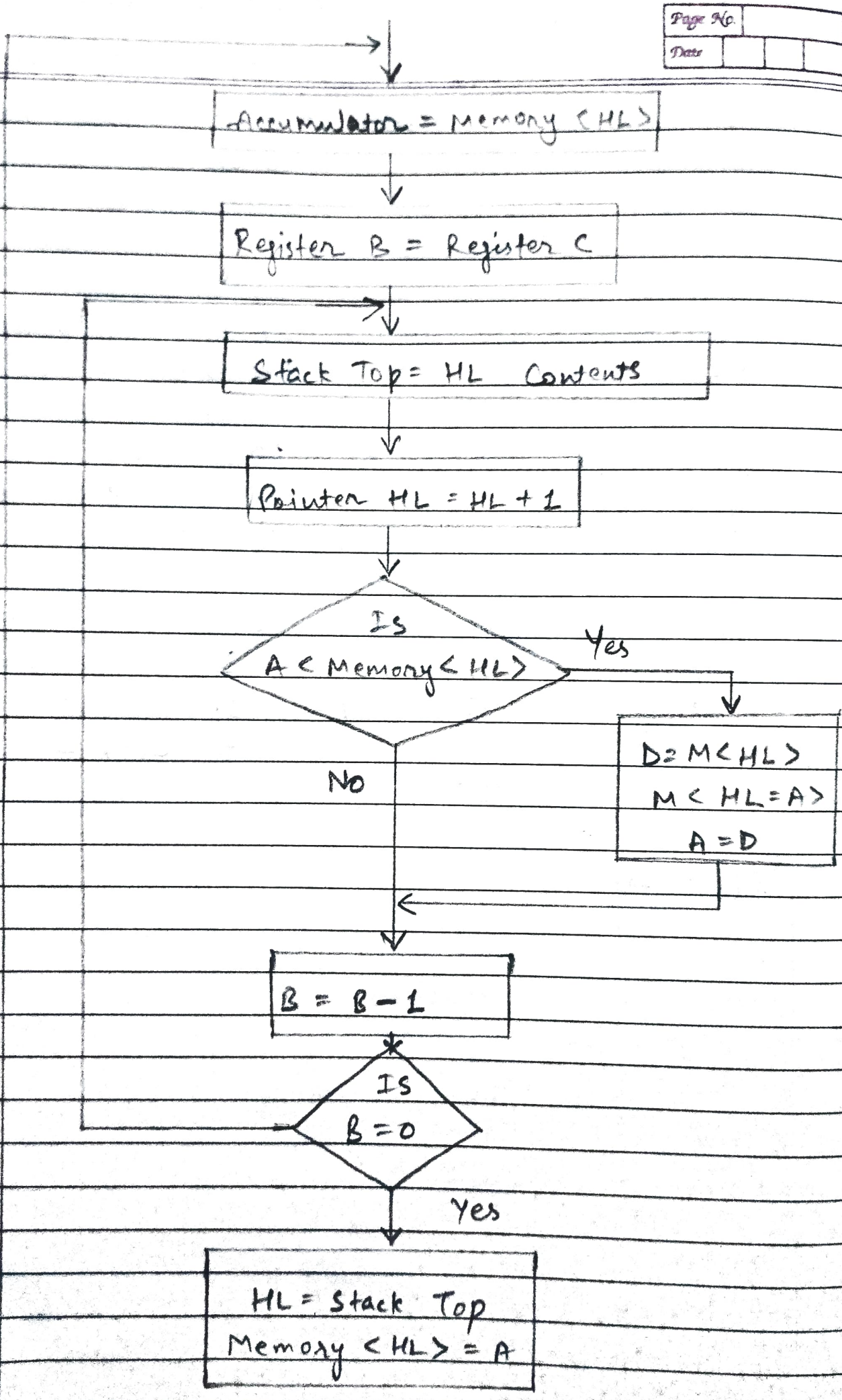
Program:

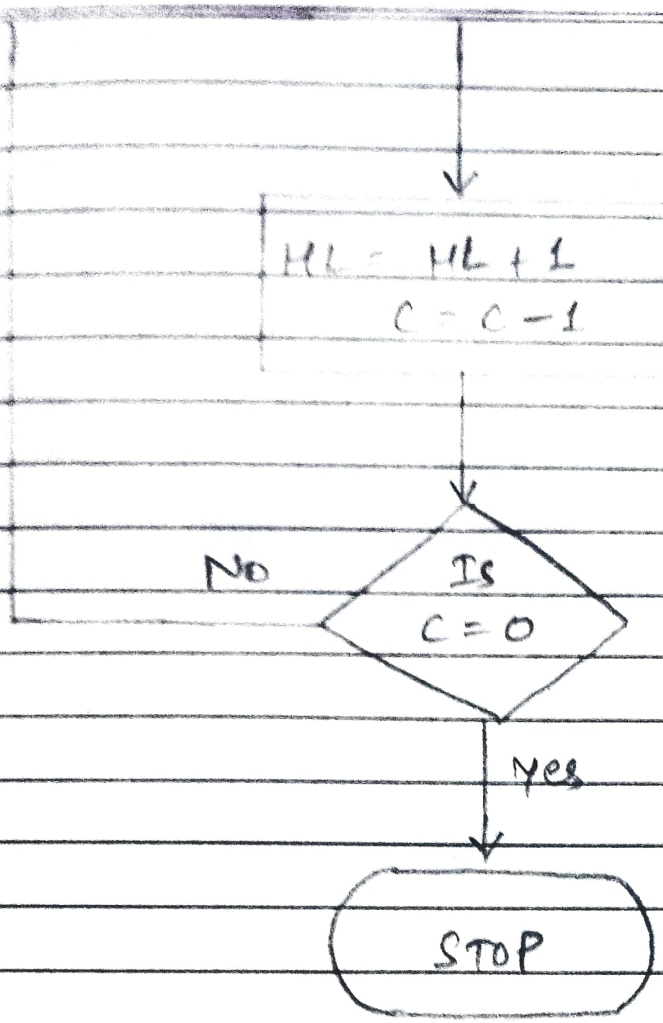
Memory Address	Machine Code	Label	Mnemonics	Operands	Comments
2000	31,00,25		LXI	SP,2500H	Initialize Stack Pointer
2003	21,50,20		LXI	H,2050H	Place length of the series
2006	4E		MOV	C,M	In register C
2007	23		INX	H	Increment H-L to point 1st memory location
2008	0D		DCR	C	C=C-1 (No. of comparison is one less)
2009	7E	LOOP2	MOV	A,M	Transfer number from memory to A.
200A	41		MOV	B,C	B=C for inner loop counter
200B	E5		PUSH	H	Save HL contents on stack
200C	23	LOOP1	INX	H	Compare next numbers in memory
200D	BE		CMP	M	with contents of A
200E	D2,14,20		JNC	AHEAD	If no carry (A>M) jump AHEAD.
2011	56		MOV	D,M	Exchange contents of memory

2012	77	MOV	M,A	and Register A using Register D
2013	7A	MOV	A,D	Register, D
2014	05 AHEAD	DCR	B	Decrement Inner Loop Counter
2015	C2,0C,20	JNZ	LOOP1	If B≠0, jump to LOOP1
2018	E1	POP	H	Get Back HL contents from stack
2019	77	MOV	M,A	Save accumulator at memory location
201A	23	INX	H	HL = HL+1 for next round
201B	0D	DCR	C	Decrement outer Loop Counter
201C	C2,09,20	JNZ	LOOP2	If not zero, jump to LOOP 2
201f	76	HLT		

Flowchart:




Accumulator = Memory <HL>
Register B = Register C
Stack Top = HL Contents
Pointer HL = HL + 1
Is A < Memory <HL>
Yes
D = M<HL>
M<HL = A>
A = D
No
B = B - 1
Is B = 0
Yes
HL = Stack Top
Memory <HL> = A


HL = HL + 1
C = C - 1
Is
C = 0
No
Yes
STOP