MICROPROCESSORS AND MICROCONTROLLERS LAB

PAPER CODE (ETEE-358)

Faculty Name : Prof. Sunil Mathur

Student name : Sarthak Bhatia

Roll No. : 07714803118

Semester : 6th

Group: I-4

Maharaja Agrasen Institute of Technology, PSP Area,

Sector – 22, Rohini, New Delhi – 110086

MICROPROCESSOR &

MICROCONTROLLER

LAB PRACTICAL

RECORD

PAPER CODE : ETEE-358

Name of the student : Sarthak Bhatia

University Roll No. : 07714803118

Branch : IT

Section/ Group : 6I - 4

PRACTICAL DETAILS

Exp. No	Experiment Name	Date of performance	Remarks	Marks(10)
1	Write a program for 8085 to add two numbers of 16 bytes each.	19/04/21		
2	Write a program for 8085 to find the largest among 10 8-bit numbers.	26/04/21		
3	Write a program for 8086 to add three 16-bit numbers.	03/05/21		
4	Write a program for 8086 to add two 10-byte numbers.	10/05/21		
5	Write a program for 8086 to arrange a series in ascending order.	24/05/21		
6	Write a program for 8086 to generate the first 8 elements of Fibonacci Series.	31/05/21		
7	Write a program for 8086 to find out the alphabet in a string series.	07/06/21		

Date

Expt. No.

Expt. Name

Page No.

EXPERIMENT-1

- **AIM-** Write a program for 8085 to add two numbers of 16 bytes each.

- **ALGORITHM-** The numbers are stored into memory at location 3501H and 3511H. One additional information is stored at location 3500 H. In this place we are storing byte count. The result is stored at location 35F0 H.

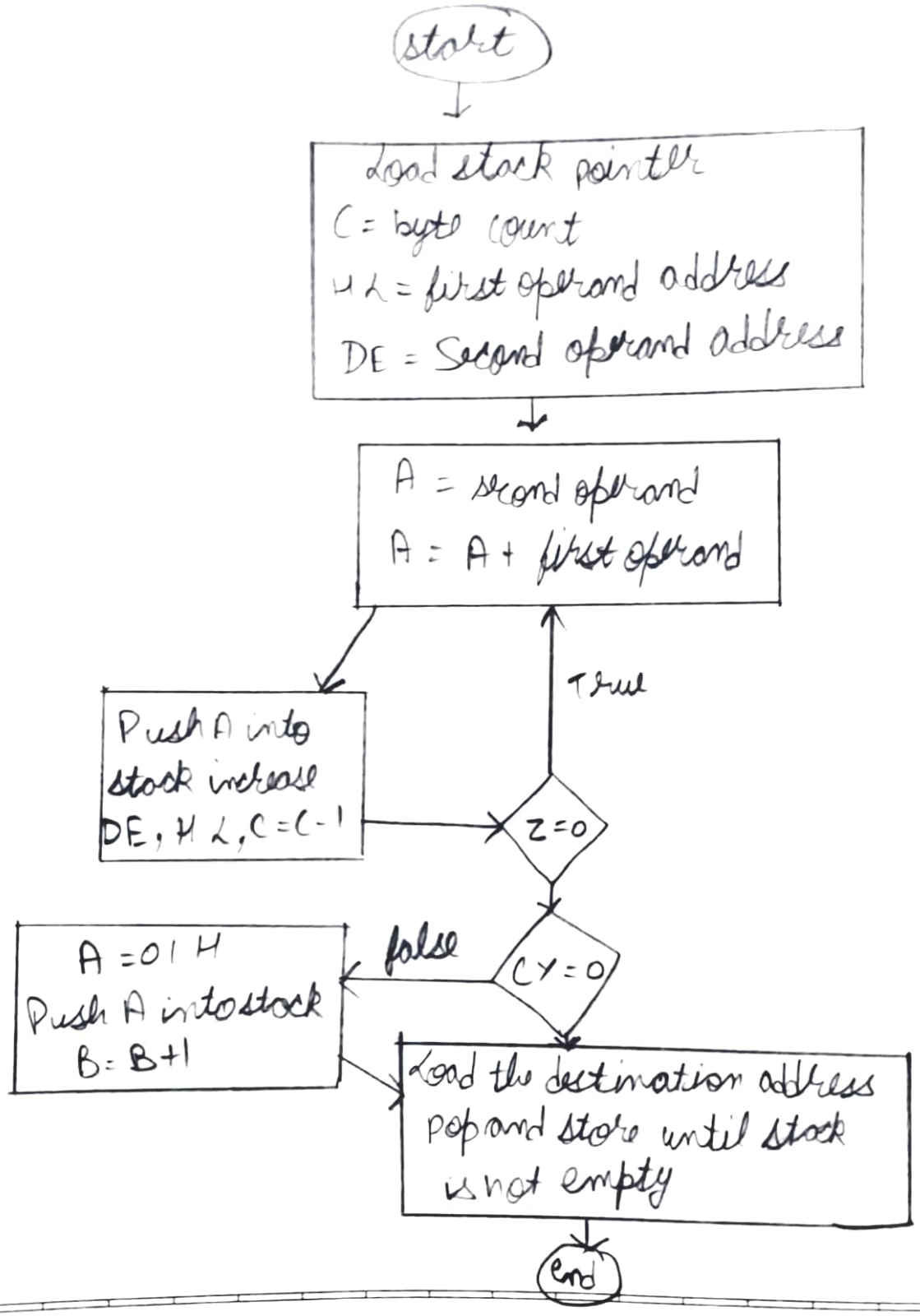
The HL pair is storing address of first operand bytes, the DE is storing address of second operand bytes. C is holding byte count. We are using stack to store the intermediate bytes of result. After completion of addition operation, we are popping from the stack and storing into the destination.

- **PROGRAM-**

	Address		Mnemonics.
1.	1000		LXI SP, 200H
2.	1003		LXI H, 3500H
3.	1006		MOV C, M
4.	1007		MVI B, 00H
5.	1009		LXI H, 3501H
6.	100C		LXI D, 3511H
7.	100F	LOOP	LDAX D

Teacher's Signature : _____

Flow Chart


start
Load stack pointer
C = byte count
HL = first operand address
DE = Second operand address
A = second operand
A = A + first operand
True
Push A into stack increase DE, HL, C = C - 1
Z = 0
false
CY = 0
A = 01 H
Push A into stack
B = B + 1
Load the destination address pop and store until stack is not empty
end

Date Expt. No.

Expt. Name Page No.

8.	1010		ADC M
9.	1011		PUSH PSW
10.	1012		INR B
11.	1013		INX H
12.	1014		INX D
13.	1015		DCR C
14.	1016		JNZ LOOP
15.	1019		JNC SKIP
16.	101C		MVI A, 01H
17.	101E		PUSH PSW
18.	101F		INR B
19.	1020	SKIP	LXI H, 8350H
20.	1023	L1	POP PSW
21.	1024		MOV M, A
22.	1025		INX H
23.	1026		DCR B
24.	1027		JNZ L1
25.	102A		RST 5

- RESULT- The input and output table is mentioned and verified. We obtained the sum of 2 16 byte numbers in input, as a 16 byte output.

→

Teacher's Signature : _____

Input

Address	Data
3501	16
3502	19
3503	68
3504	12
3505	65
3506	88
3507	25
3508	17
3509	20
350A	23
350B	16
350C	12
350D	21
350E	35
350F	25
3510	18
3511	41
3512	13
3513	49
3514	56
3515	62
3516	33
3517	29
3518	67
3519	72
351A	29
351B	44
351C	22
351D	11
351E	69
351F	58
3520	47
3521	19

Output

Address	Data
35F0	00
35F1	51
35F2	5F
35F3	7D
35F4	9E
35F5	3B
35F6	34
35F7	5A
35F8	8C
35F9	92
35FA	7E
35FB	4E
35FC	BB
35FD	E7
35FE	68
35FF	81
3600	2C

Date

Expt. No.

Expt. Name

Page No.

VIVA QUESTIONS

Q1 What is the function of HLT key of 8085 microprocessor.

→ HLT is a mnemonic which stands for 'Halt' the microprocessor instruction. Using these particular instruction, 8085 microprocessor enter into halt state.

Q2 How can you store data at a memory location?

→ We can store data at memory location using LDA or LXI command.

Q3 What flags are available in 8085 microprocessor?

→ There are 5 flags:-

1. Carry flag.
2. Auxiliary flag.
3. Sign flag.
4. Parity flag
5. Zero flag.

Q4 What is the function of LHLD 8501H instruction.

→ It initialise HL register pair from 8501H and 8502H memory location.

Q5 How many data lines and address lines are present in 8085 microprocessor?

→ There are 8 data lines and 16 address lines.

Teacher's Signature :

Date Expt. No.

Expt. Name Page No.

EXPERIMENT-2

- AIM - Write a program for 8085 to find the largest among 10 8-bit numbers.

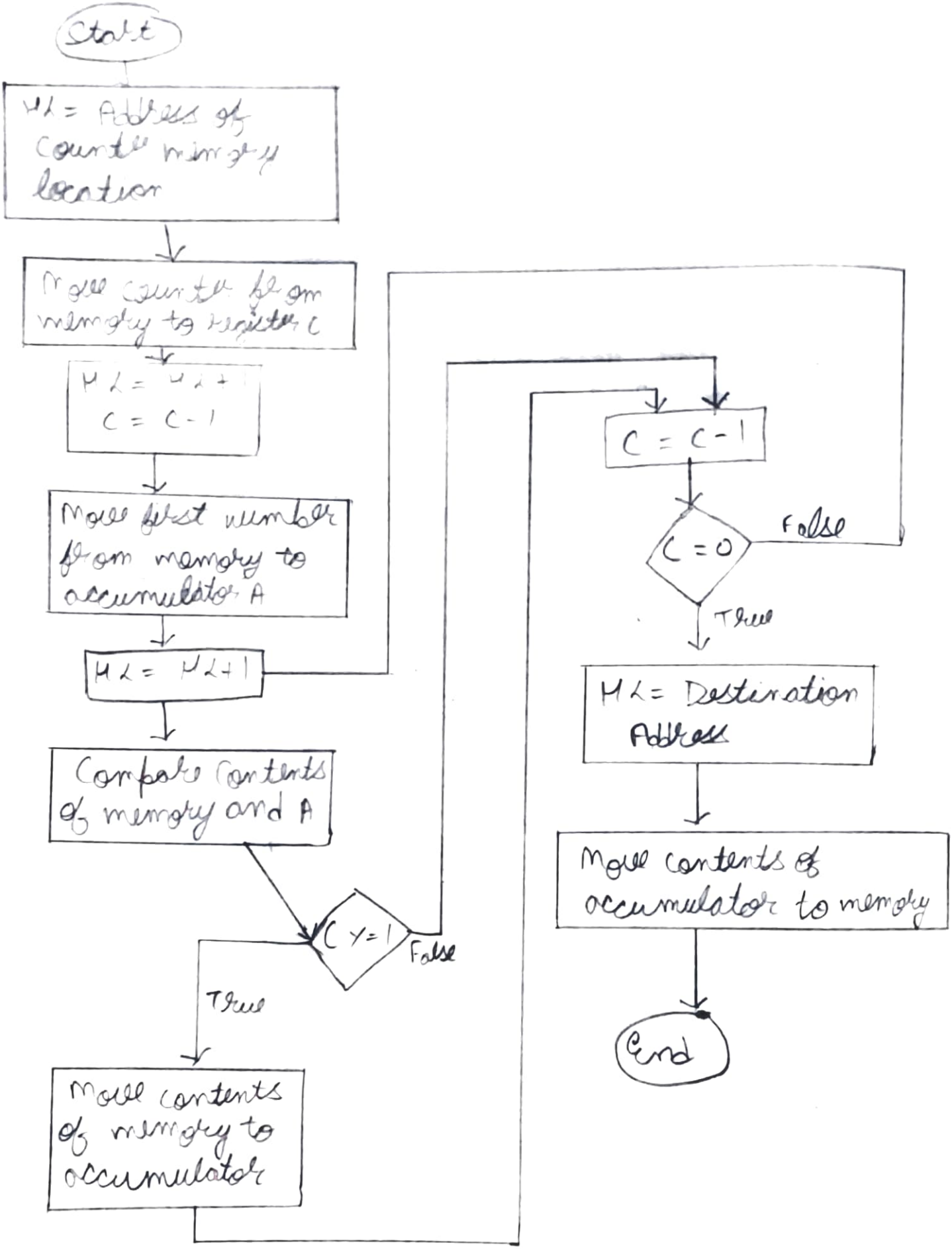
- ALGORITHM - In this program data is stored from location 1000H. Memory location 1000H contains size of array of elements i.e. 10. Data elements are stored starting from location 1001H. After execution of the program we will store the result at 2000H memory location.

In this program, we will start by ~~comparing~~ storing first number in accumulator and then we will compare data in accumulator with all other elements one by one and keeping storing maximum ~~at~~ in accumulator. In each iteration the counter of size is also decreased. At the end, we will store accumulator data in 2000H.

- PROGRAM

Address	Mnemonics
4000	LXI H, 1000
4003	MOV C, M
4004	INR H
4005	~~MOV~~ DCR C

Teacher's Signature : _____


Start
HL = Address of counter memory location
Move counter from memory to register C
HL = HL + 1
C = C - 1
Move first number from memory to accumulator A
HL = HL + 1
Compare contents of memory and A
CY = 1
False
True
Move contents of memory to accumulator
C = C - 1
C = 0
False
True
HL = Destination Address
Move contents of accumulator to memory
End

Date

Expt. No.

Expt. Name

Page No.

4006		MOV B, M
4007	LOOP	INR H
4008		MOV A, M
4009		CMP B
400A		JNC DO
400D		MOV B, A
400E	DO	DCR C
400F		JNC LOOP
4012		LXI H, 2000H
4015		MOV M, B
4016		RST 5

- VIVA QUESTIONS

Q1 How many data lines and address lines are in 8085 microprocessor?

→ There are 8 data lines and 16 Address lines in 8085.

Q2 Give examples of one, two and three byte instructions?

→ 1 byte MOV A, M

2 byte MVI B, 00H

3 byte LXI H, 8000H

Q3 What is a microcontroller?

→ It is a single chip on which a microprocessor, memory and i/o signal lines are fabricated.

→

Teacher's Signature : _____

Date

Expt. No.

Expt. Name

Page No.

Q4 What is the difference between SUB M and CMP M ? Both perform A - M.

→ Sub will store the result in A and affect the flags. CMP will not store the result, it will only affect the flags.

Q5 What is the significance of accumulator in 8085? A is the accumulator.

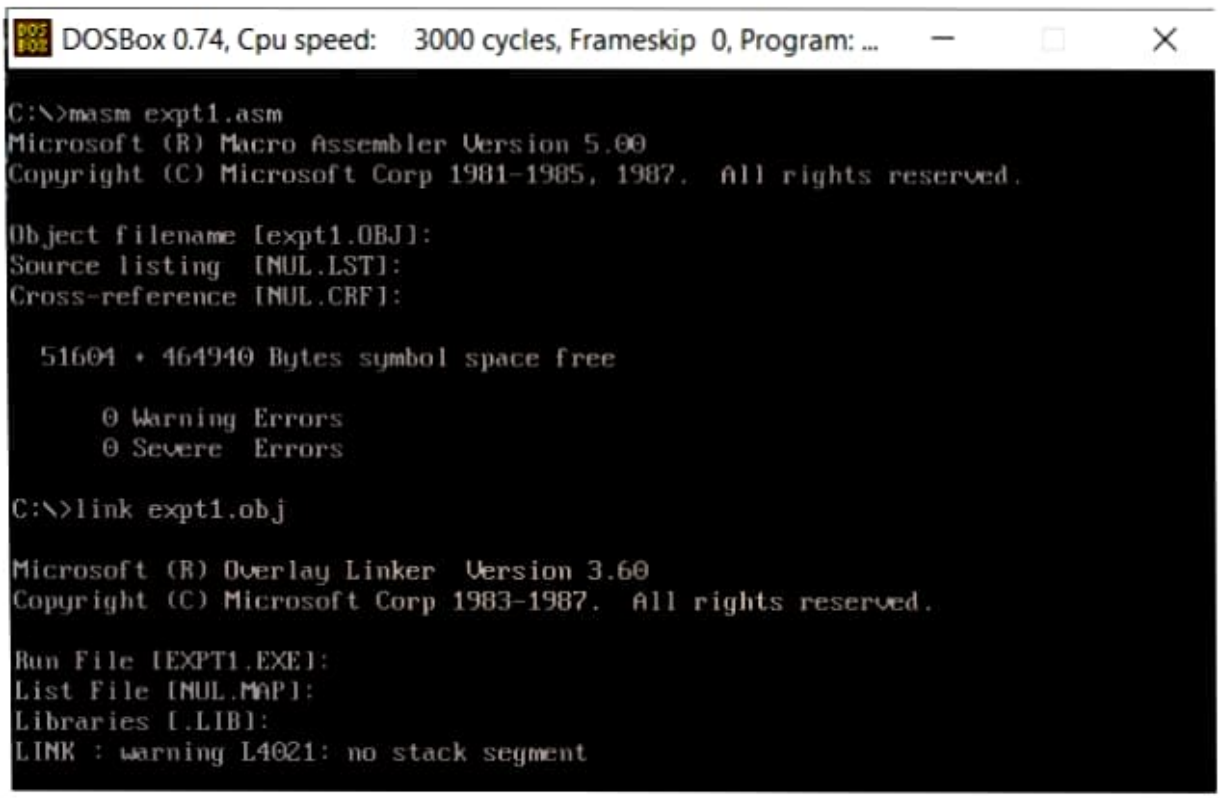
→ It holds the first operand and the result in arithmetic and logical operations.

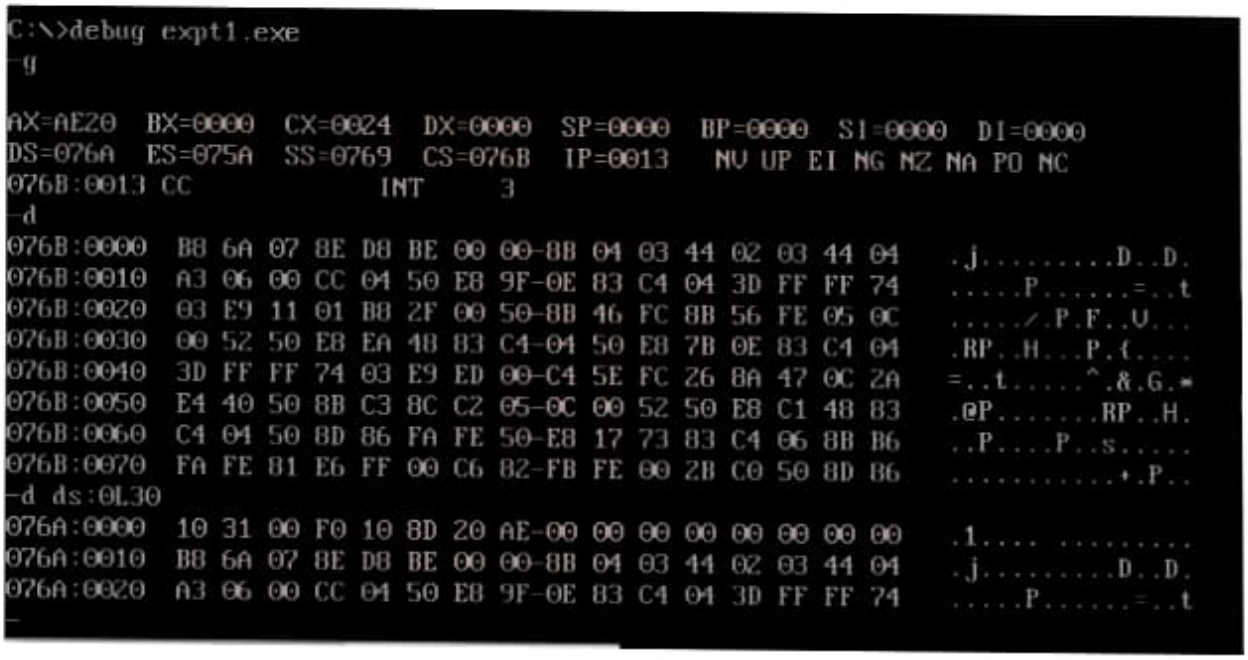
Teacher's Signature : _____

Experiment 3

Aim : To add three 16 bit numbers in 8086

```
SARTHAK SEGMENT
MEM1 DW 3110H, 0F000H, 8D10H
RES DW ?
SARTHAK ENDS

BHATIA SEGMENT
ASSUME DS: SARTHAK, CS: BHATIA
START:
MOV AX, SARTHAK
MOV DS, AX
MOV SI, OFFSET MEM1
MOV AX, [SI]
ADD AX, [SI+2]
ADD AX, [SI+4]
MOV RES, AX
INT 03H
BHATIA ENDS
END START
END
```

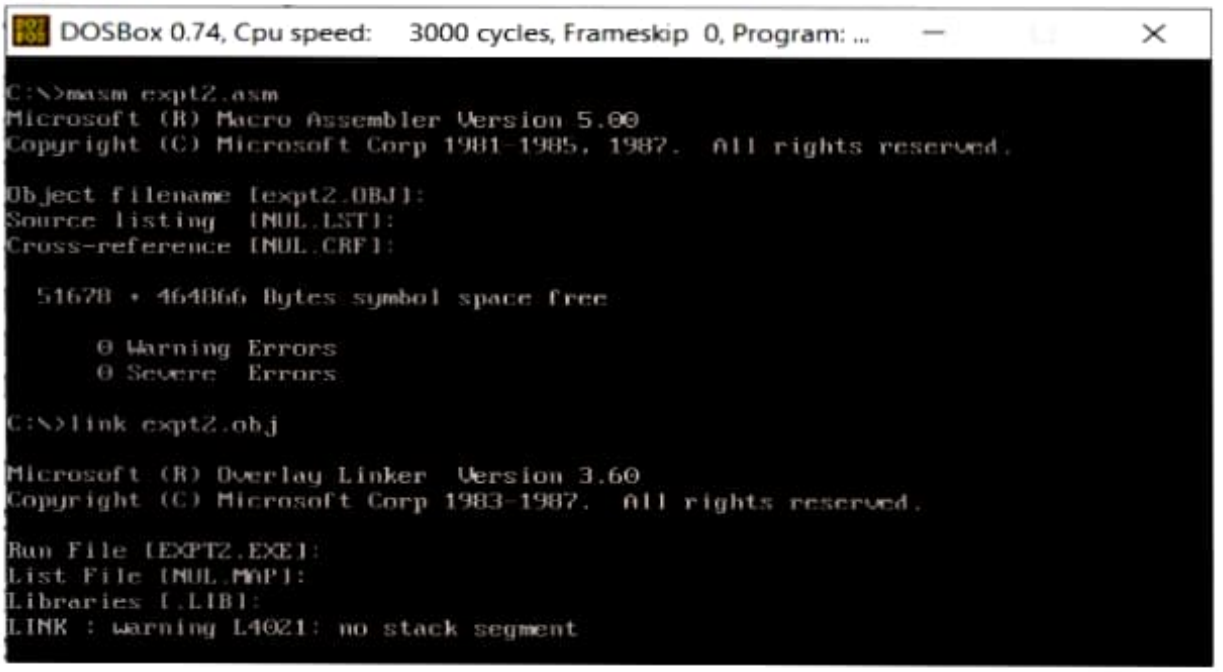


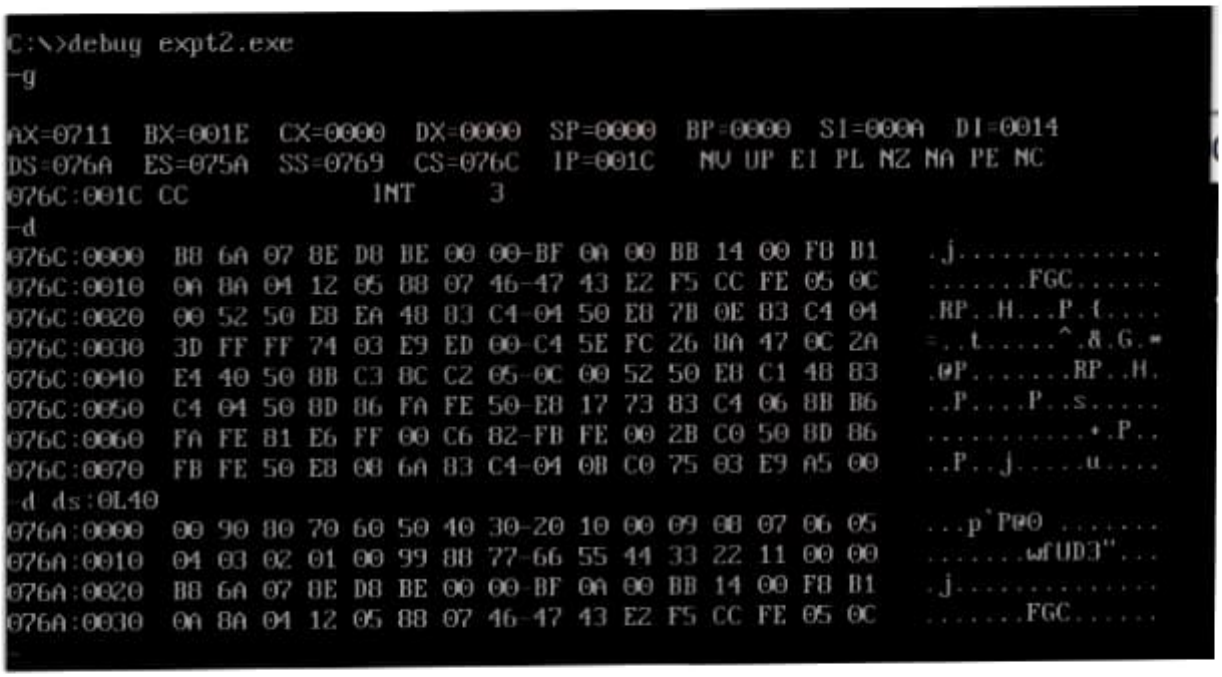

C:\>debug expt1.exe
-g
AX=AE20 BX=0000 CX=0024 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0013 NV UP EI NG NZ NA PO NC
076B:0013 CC INT 3
-d
076B:0000 B8 6A 07 8E D8 BE 00 00-8B 04 03 44 02 03 44 04 .j.....D..D.
076B:0010 A3 06 00 CC 04 50 E8 9F-0E 83 C4 04 3D FF FF 74 .....P.....=..t
076B:0020 03 E9 11 01 B8 2F 00 50-8B 46 FC 8B 56 FE 05 0C ...../.P.F..V...
076B:0030 00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04 .RP..H...P.{....
076B:0040 3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A =..t.....^.&.G.*
076B:0050 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83 .@P.....RP..H.
076B:0060 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6 ..P....P..s.....
076B:0070 FA FE 81 E6 FF 00 C6 82-FB FE 00 2B C0 50 8D 86 .....+.P..
-d ds:0L30
076A:0000 10 31 00 F0 10 8D 20 AE-00 00 00 00 00 00 00 00 .1.... .....
076A:0010 B8 6A 07 8E D8 BE 00 00-8B 04 03 44 02 03 44 04 .j.....D..D.
076A:0020 A3 06 00 CC 04 50 E8 9F-0E 83 C4 04 3D FF FF 74 .....P.....=..t
-

Experiment 4

Aim : To add two 10 byte numbers in 8086

```
SARTHAK SEGMENT
MEM1 DB 00H,90H,80H,70H,60H,50H,40H,30H,20H,10H
MEM2 DB 00H,09H,08H,07H,06H,05H,04H,03H,02H,01H
RES DB 10 DUP (?)
SARTHAK ENDS

BHATIA SEGMENT
ASSUME DS: SARTHAK, CS: BHATIA
START:
    MOV AX,SARTHAK
    MOV DS,AX
    MOV SI,OFFSET MEM1
    MOV DI,OFFSET MEM2
    MOV BX,OFFSET RES
    CLC
    MOV CL,0AH
UP: MOV AL,[SI]
    ADC AL,[DI]
    MOV [BX],AL
    INC SI
    INC DI
    INC BX
    LOOP UP
   INT 03H
BHATIA ENDS
END START
END
```




C:\>debug expt2.exe
-g
AX=0711 BX=001E CX=0000 DX=0000 SP=0000 BP=0000 SI=000A DI=0014
DS=076A ES=075A SS=0769 CS=076C IP=001C NV UP EI PL NZ NA PE NC
076C:001C CC INT 3
-d
076C:0000 B8 6A 07 8E D8 BE 00 00-BF 0A 00 BB 14 00 FB B1 .j..............
076C:0010 0A 8A 04 12 05 88 07 46-47 43 E2 F5 CC FE 05 0C ....FGC......
076C:0020 00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04 .RP..H...P.{....
076C:0030 3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A =..t.....^.&.G.*
076C:0040 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83 .@P.......RP..H.
076C:0050 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6 ..P....P..s.....
076C:0060 FA FE 81 E6 FF 00 C6 82-FB FE 00 2B C0 50 8D 86 ....+.P..
076C:0070 FB FE 50 E8 08 6A 83 C4-04 0B C0 75 03 E9 A5 00 ..P..j.....u....
-d ds:0L40
076A:0000 00 90 80 70 60 50 40 30-20 10 00 09 08 07 06 05 ...p`P@0 ....
076A:0010 04 03 02 01 00 99 88 77-66 55 44 33 22 11 00 00 ....wfUD3"...
076A:0020 B8 6A 07 8E D8 BE 00 00-BF 0A 00 BB 14 00 FB B1 .j..............
076A:0030 0A 8A 04 12 05 88 07 46-47 43 E2 F5 CC FE 05 0C ....FGC......

EXPERIMENT-5

Name- Sarthak Bhatia

Group-I4

Roll No.- 07714803118

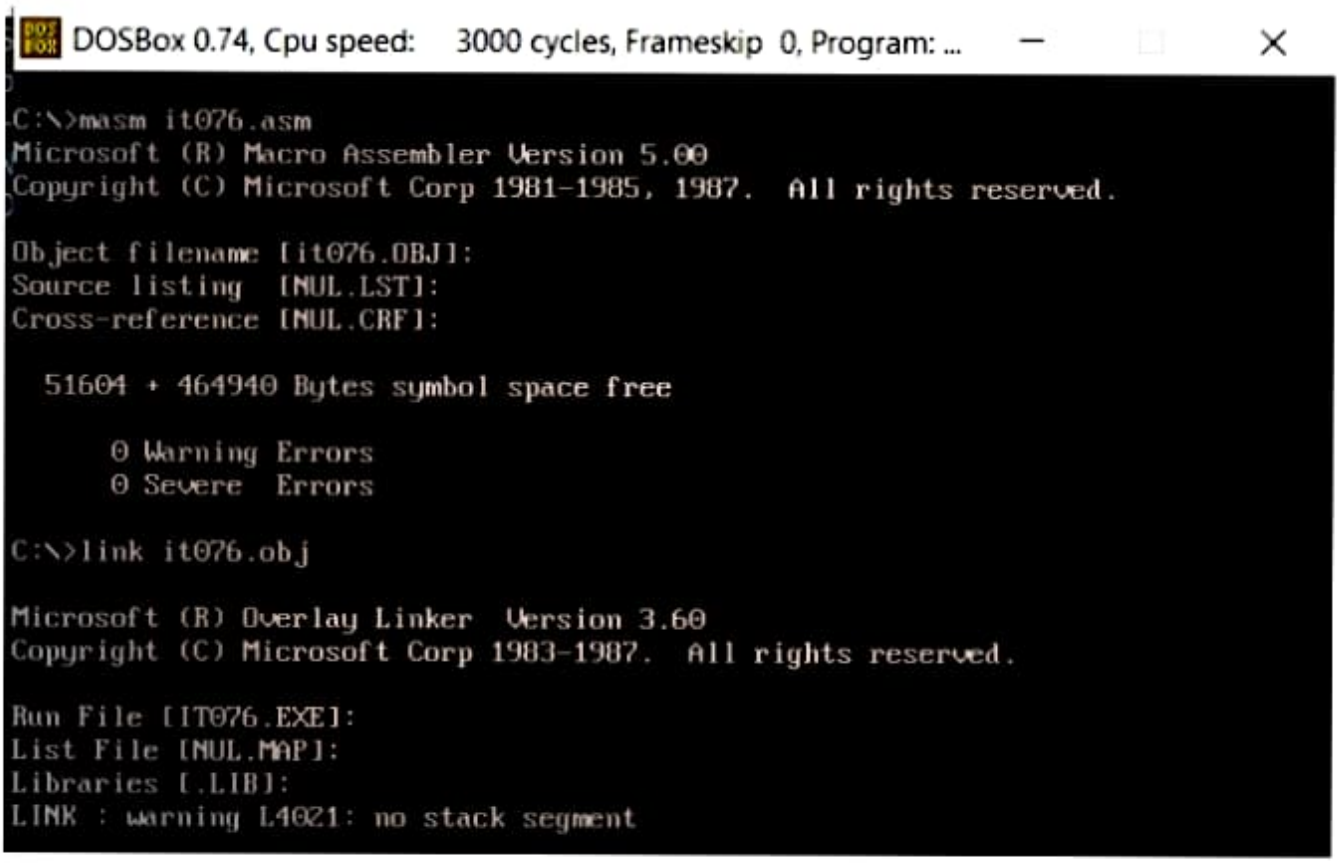
AIM- Write a program to arrange a series in ascending order in 8086 microprocessor.

PROGRAM-

```
SARTHAK SEGMENT
STRING1 DB 99H,15H,63H,54H,32H
SARTHAK ENDS

BHATIA SEGMENT
ASSUME CS:BHATIA , DS:SARTHAK
START:  MOV AX,SARTHAK
        MOV DS,AX
        MOV CH,04H
UP2:    MOV CL,04H
        LEA SI,STRING1
UP1:    MOV AL,[SI]
        MOV BL,[SI+1]
        CMP AL,BL
        JC DOWN
        MOV DL,[SI+1]
```

```
        XCHG [SI],DL
        MOV [SI+1],DL
DOWN:   INC SI
        DEC CL
        JNZ UP1
        DEC CH
        JNZ UP2
        INT 03H
BHATIA ENDS
END START
```

OUTPUT-


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
C:\>masm it076.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.
Object filename [it076.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
51604 + 464940 Bytes symbol space free
0 Warning Errors
0 Severe Errors
C:\>link it076.obj
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.
Run File [IT076.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment


C:\>debug it076.exe
-g
AX=0763 BX=0099 CX=0000 DX=0054 SP=0000 BP=0000 SI=0004 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0027 NV UP EI PL ZR NA PE CY
076B:0027 CC INT 3
-d
076B:0000 B8 6A 07 8E D8 B5 04 B1-04 8D 36 00 00 8A 04 8A .j.....6.....
076B:0010 5C 01 3A C3 72 08 8A 54-01 86 14 88 54 01 46 FE \.:.r..T....T.F.
076B:0020 C9 75 EA FE CD 75 E0 CC-8B 46 FC 8B 56 FE 05 0C .u...u...F..V...
076B:0030 00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04 .RP..H...P.{....
076B:0040 3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A =..t.....^.&.G.*
076B:0050 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83 .@P.......RP..H.
076B:0060 C4 04 50 8D 86 FA FE 50-EB 17 73 83 C4 06 8B B6 ..P....P..s.....
076B:0070 FA FE 81 E6 FF 00 C6 82-FB FE 00 2B C0 50 8D 86 ....+.P..
-d ds:0L30
076A:0000 15 32 54 63 99 00 00 00-00 00 00 00 00 00 00 00 .2Tc............
076A:0010 B8 6A 07 8E D8 B5 04 B1-04 8D 36 00 00 8A 04 8A .j.....6.....
076A:0020 5C 01 3A C3 72 08 8A 54-01 86 14 88 54 01 46 FE \.:.r..T....T.F.

EXPERIMENT-6

Name-Sarthak Bhatia
Roll No.- 07714803118
Group-I4

AIM- Write a program to generate the 1st 8 elements of the Fibonacci series.

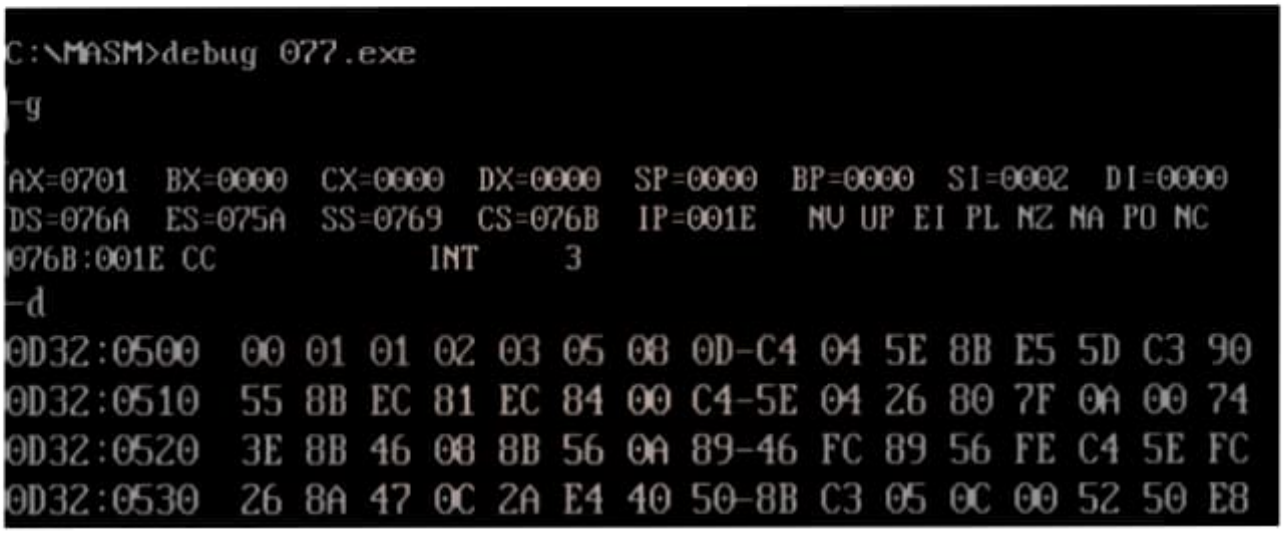
PROGRAM-

```
SARTHAK BHATIA
MEM1 DB 08H
RES DB 8 DUP (?)
SARTHAK ENDS

BHATIA SEGMENT
ASSUME CS:BHATIA , DS:SARTHAK

START:  MOV AX,SARTHAK
        MOV DS,AX
        MOV SI, OFFSET(RES)
        MOV AL , 00H
        MOV [SI],AL
        ADD SI, 01H
        ADD AL, 01H
        MOV [SI],AL
        MOV CX , OFFSET(MEM1)
        SUB CX,0002H
```

```
    L1 : MOV AL, [SI-1]
         ADD AL,[SI]
         INC SI
         MOV [SI],AL
         LOOP L1
         INT 03H
BHATIA ENDS
END START
END
```

OUTPUT-

```
C:\MASM>masm 077.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [077.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51612 + 464932 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\MASM>link 077.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [077.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment
```


C:\MASM>debug 077.exe
-g
AX=0701 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0002 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=001E NV UP EI PL NZ NA PO NC
076B:001E CC INT 3
-d
0D32:0500 00 01 01 02 03 05 08 0D-C4 04 5E 8B E5 5D C3 90
0D32:0510 55 8B EC 81 EC 84 00 C4-5E 04 26 80 7F 0A 00 74
0D32:0520 3E 8B 46 08 8B 56 0A 89-46 FC 89 56 FE C4 5E FC
0D32:0530 26 8A 47 0C 2A E4 40 50-8B C3 05 0C 00 52 50 E8

Experiment 7

Aim : Write a program to find out an alphabet in a string series

```
SARTHAK SEGMENT
        STR1 DB "abadegehlo"
        A DB 0H
        MSG1 DB 10,13,"COUNT OF A's IS : $"
SARTHAK ENDS

DISPLAY MACRO MSG
        MOV AH,9
        LEA DX,MSG
        INT 21H
ENDM

BHATIA SEGMENT
        ASSUME CS:SARTHAK,DS:BHATIA
        START:
                MOV AX,SARTHAK
                MOV DS,AX
                LEA SI,STR1
                MOV CX,10

        CHECK:
                MOV AL,[SI]
                CMP AL,'A'
                JNE N1
                INC A

        N1:     CMP AL,'a'
                JNE N2
                INC A

        N2: INC SI
                LOOP CHECK

        DISPLAY MSG1
```

```
        MOV DL,A
        ADD DL,30H
        MOV AH,2
        INT 21H

        MOV AH,4CH
        INT 21H
BHATIA ENDS
END START
```

```
C:\MASM>masm 077expt7.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [077expt7.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51622 + 464922 Bytes symbol space free

       0 Warning Errors
       0 Severe  Errors

C:\MASM>link 077expt7.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [077EXPT7.EXE]:
```

```
C:\MASM>debug 077expt7.exe
-g_

COUNT OF A's IS : 2
Program terminated normally
-_
```