MicroProcessor Lab

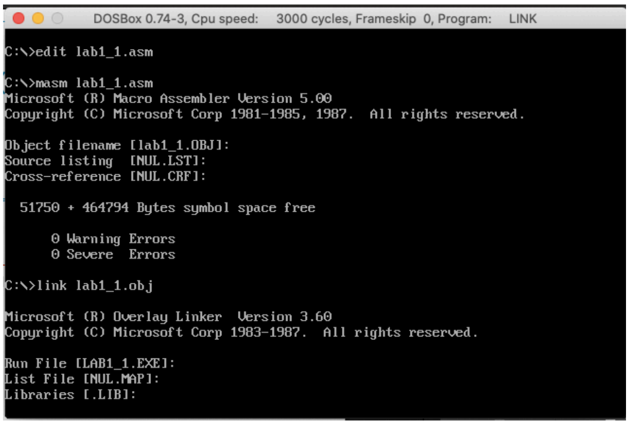
Aadhitya Swarnesh

- 3 June 2020

LAB - 1

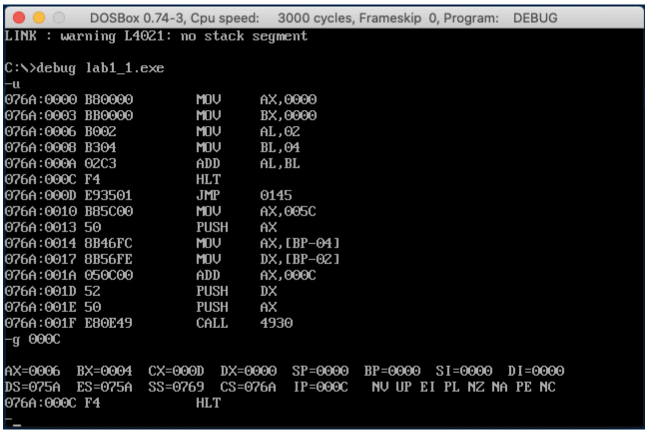
Aim :- Exploring Different addressing modes and addition of two numbers

```
ASSUME CS:CODE
CODE SEGMENT
START:
    mov al, 02H
    mov bl, 04H
    add al, bl
    hlt
CODE ENDS
END START
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: LINK

```
C:\>edit lab1_1.asm

C:\>masm lab1_1.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab1_1.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51750 + 464794 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab1_1.obj

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

Run File [LAB1_1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

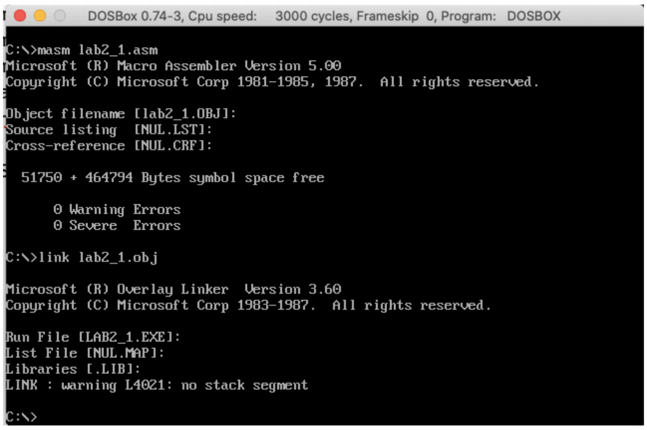
```
LINK : warning L4021: no stack segment

C:\>debug lab1_1.exe
-u
076A:0000 B80000        MOV     AX,0000
076A:0003 BB0000        MOV     BX,0000
076A:0006 B002          MOV     AL,02
076A:0008 B304          MOV     BL,04
076A:000A 02C3          ADD     AL,BL
076A:000C F4            HLT
076A:000D E93501        JMP     0145
076A:0010 B85C00        MOV     AX,005C
076A:0013 50            PUSH    AX
076A:0014 8B46FC        MOV     AX,[BP-04]
076A:0017 8B56FE        MOV     DX,[BP-02]
076A:001A 050C00        ADD     AX,000C
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
-g 000C

AX=0006  BX=0004  CX=000D  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000C   NV UP EI PL NZ NA PE NC
076A:000C F4            HLT
-_
```

LAB - 2

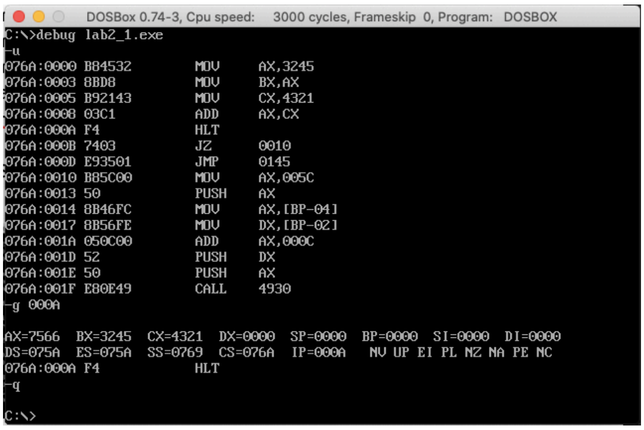
Task 1

Aim :- Addition of two 16 bit numbers using 16 bit registers.

```
CODE SEGMENT
ASSUME cs:code
START:
    mov ax, 3245H
    mov bx, ax
    mov cx, 4321H
    add ax, cx
    hlt
CODE ENDS
END START
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>masm lab2_1.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab2_1.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51750 + 464794 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab2_1.obj

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

Run File [LAB2_1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>
```

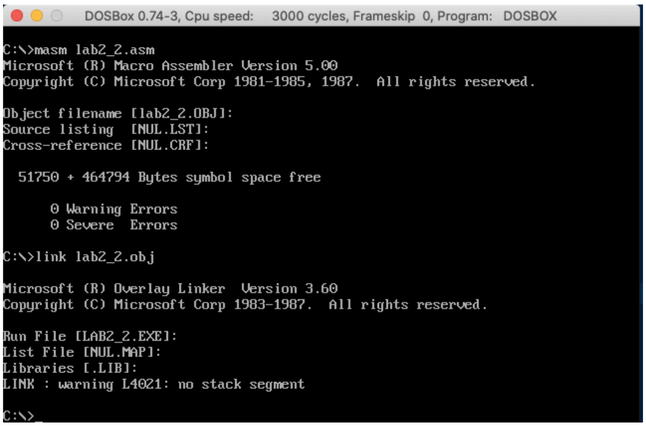

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>debug lab2_1.exe
-u
076A:0000 B84532        MOV     AX,3245
076A:0003 8BD8          MOV     BX,AX
076A:0005 B92143        MOV     CX,4321
076A:0008 03C1          ADD     AX,CX
076A:000A F4            HLT
076A:000B 7403          JZ      0010
076A:000D E93501        JMP     0145
076A:0010 B85C00        MOV     AX,005C
076A:0013 50            PUSH    AX
076A:0014 8B46FC        MOV     AX,[BP-04]
076A:0017 8B56FE        MOV     DX,[BP-02]
076A:001A 050C00        ADD     AX,000C
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
-g 000A

AX=7566  BX=3245  CX=4321  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000A   NV UP EI PL NZ NA PE NC
076A:000A F4            HLT
-q

C:\>
```

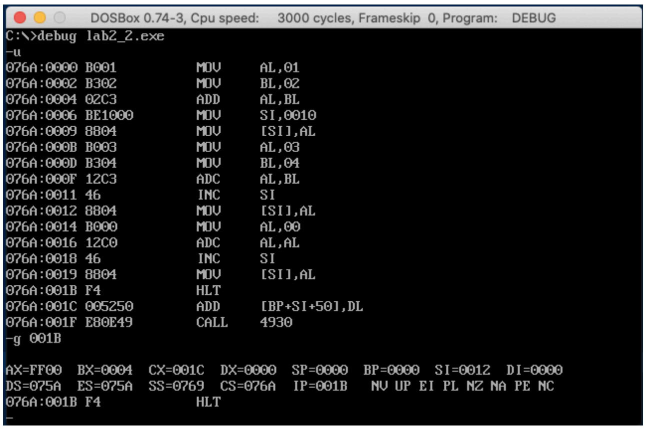
Task 2

Aim :- Addition of two 16 bit numbers using 8 bit registers.

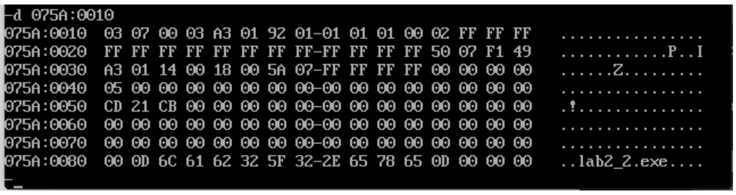
```
CODE SEGMENT
ASSUME cs:code
START:
    mov al, 01H
    mov bl, 02H
    add al, bl
    mov si, 10H
    mov [si], al
    mov al, 03H
    mov bl, 04H
    adc al, bl
    inc si
    mov [si], al
    mov al, 00H
    adc al, al
    inc si
    mov [si], al
    hlt
CODE ENDS
END START
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>masm lab2_2.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab2_2.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51750 + 464794 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab2_2.obj

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

Run File [LAB2_2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>_
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
C:\>debug lab2_2.exe
-u
076A:0000 B001          MOV     AL,01
076A:0002 B302          MOV     BL,02
076A:0004 02C3          ADD     AL,BL
076A:0006 BE1000        MOV     SI,0010
076A:0009 8804          MOV     [SI],AL
076A:000B B003          MOV     AL,03
076A:000D B304          MOV     BL,04
076A:000F 12C3          ADC     AL,BL
076A:0011 46            INC     SI
076A:0012 8804          MOV     [SI],AL
076A:0014 B000          MOV     AL,00
076A:0016 12C0          ADC     AL,AL
076A:0018 46            INC     SI
076A:0019 8804          MOV     [SI],AL
076A:001B F4            HLT
076A:001C 005250        ADD     [BP+SI+50],DL
076A:001F E80E49        CALL    4930
-g 001B

AX=FF00  BX=0004  CX=001C  DX=0000  SP=0000  BP=0000  SI=0012  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=001B   NV UP EI PL NZ NA PE NC
076A:001B F4            HLT
-
```

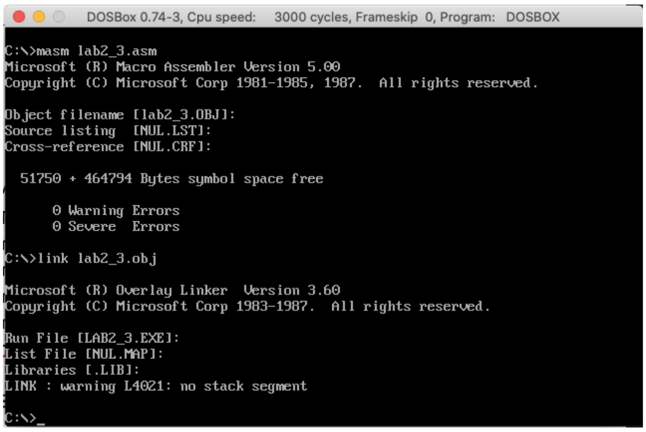


```
-d 075A:0010
075A:0010  03 07 00 03 A3 01 92 01-01 01 01 00 02 FF FF FF   ................
075A:0020  FF FF FF FF FF FF FF FF-FF FF FF FF 50 07 F1 49   ............P..I
075A:0030  A3 01 14 00 18 00 5A 07-FF FF FF FF 00 00 00 00   ......Z.........
075A:0040  05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
075A:0050  CD 21 CB 00 00 00 00 00-00 00 00 00 00 00 00 00   .!..............
075A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
075A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
075A:0080  00 0D 6C 61 62 32 5F 32-2E 65 78 65 0D 00 00 00   ..lab2_2.exe....
-_
```

Task 3

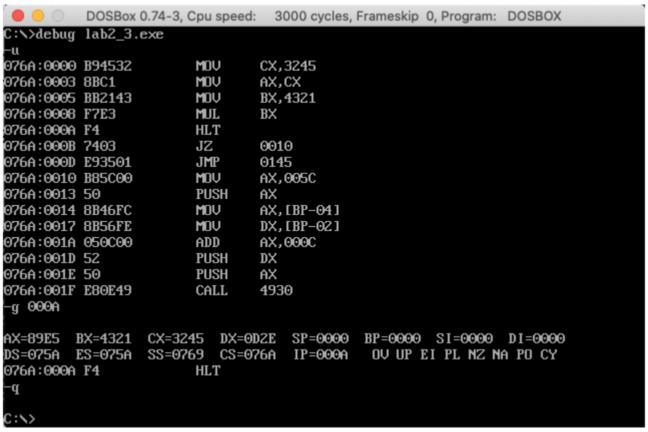
Aim :- Multiplication of two 16 bit numbers.

```
CODE SEGMENT
ASSUME cs:code
START:
    mov cx, 3245H
    mov ax, cx
    mov bx, 4321H
    mul bx
    hlt
CODE ENDS
END START
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>masm lab2_3.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab2_3.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51750 + 464794 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab2_3.obj

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

Run File [LAB2_3.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>_
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

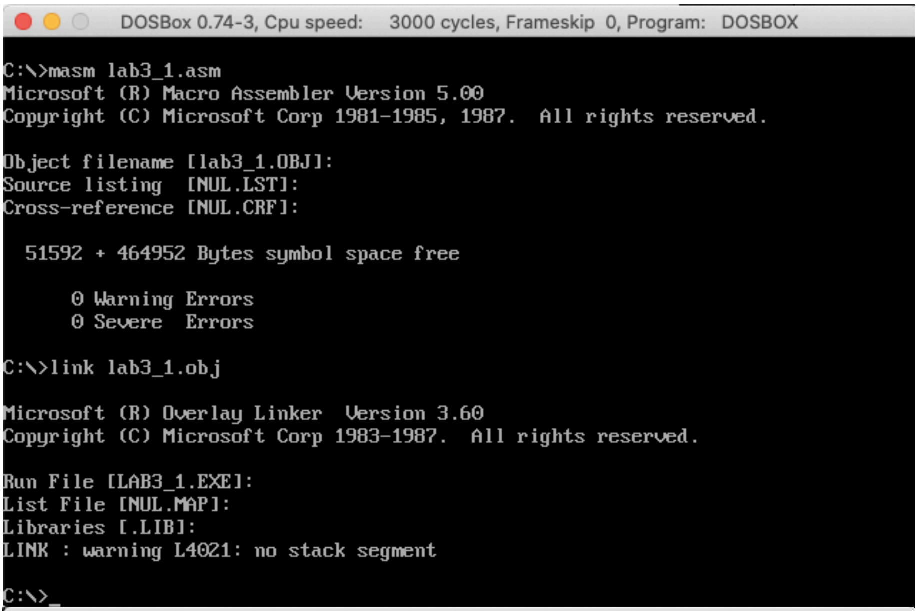
```
C:\>debug lab2_3.exe
-u
076A:0000 B94532        MOV     CX,3245
076A:0003 8BC1          MOV     AX,CX
076A:0005 BB2143        MOV     BX,4321
076A:0008 F7E3          MUL     BX
076A:000A F4            HLT
076A:000B 7403          JZ      0010
076A:000D E93501        JMP     0145
076A:0010 B85C00        MOV     AX,005C
076A:0013 50            PUSH    AX
076A:0014 8B46FC        MOV     AX,[BP-04]
076A:0017 8B56FE        MOV     DX,[BP-02]
076A:001A 050C00        ADD     AX,000C
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
-g 000A

AX=89E5  BX=4321  CX=3245  DX=0D2E  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000A   OV UP EI PL NZ NA PO CY
076A:000A F4            HLT
-q

C:\>
```

LAB - 3

Task 1

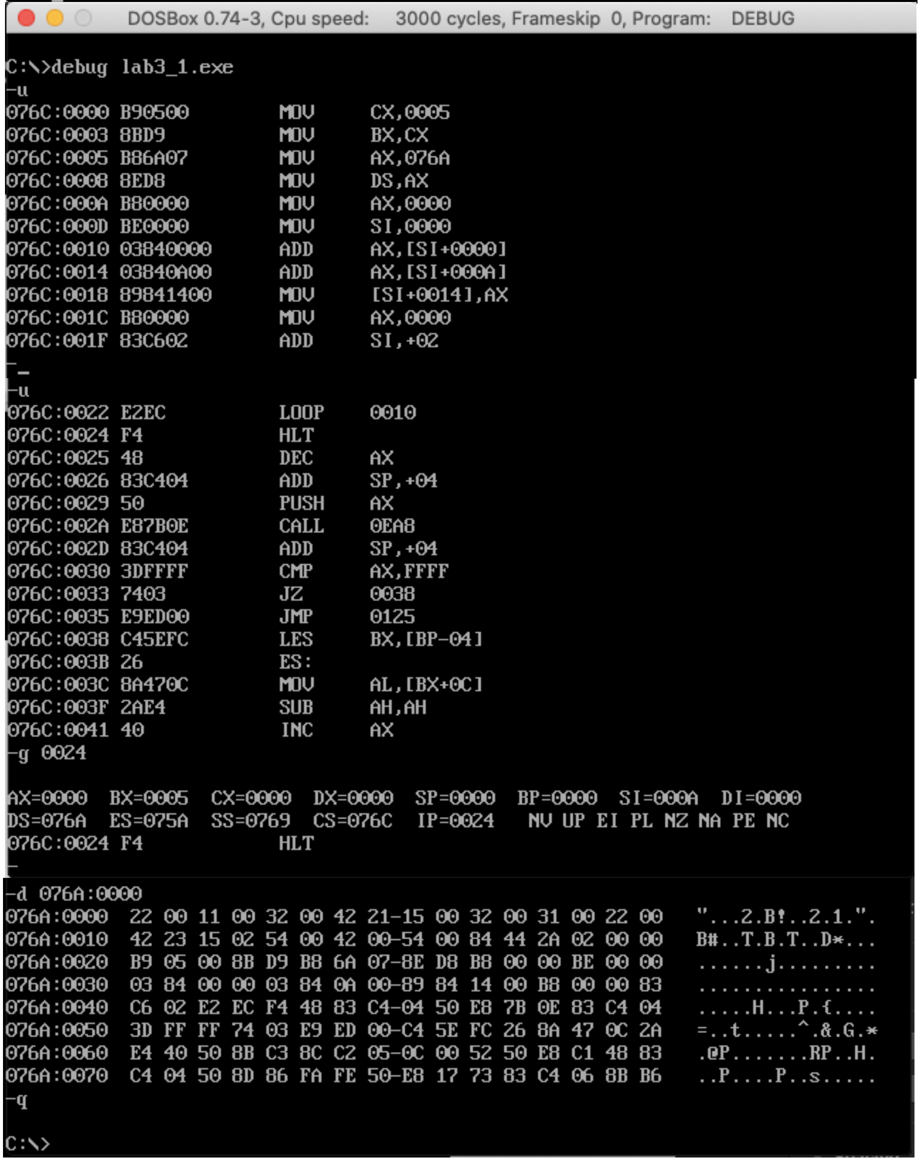
Aim :- Addition of the elements of two arrays.

```
DATA SEGMENT
    MAT1 dw 0022H, 0011H, 0032H, 2142H, 0015H
    MAT2 dw 0032H, 0031H, 0022H, 2342H, 0215H
    RESMAT dw 5 dup(0)
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:
    mov cx, 05H
    mov bx, cx
    mov ax, DATA
    mov ds, ax
    mov ax, 0000H
    mov si, 00H
    RPT:
        add ax, MAT1[si]
        add ax, MAT2[si]
        mov RESMAT[si], ax
        mov ax, 00H
        add si, 02H
    LOOP RPT
    hlt
CODE ENDS
END START
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>masm lab3_1.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab3_1.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51592 + 464952 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab3_1.obj

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

Run File [LAB3_1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>_
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

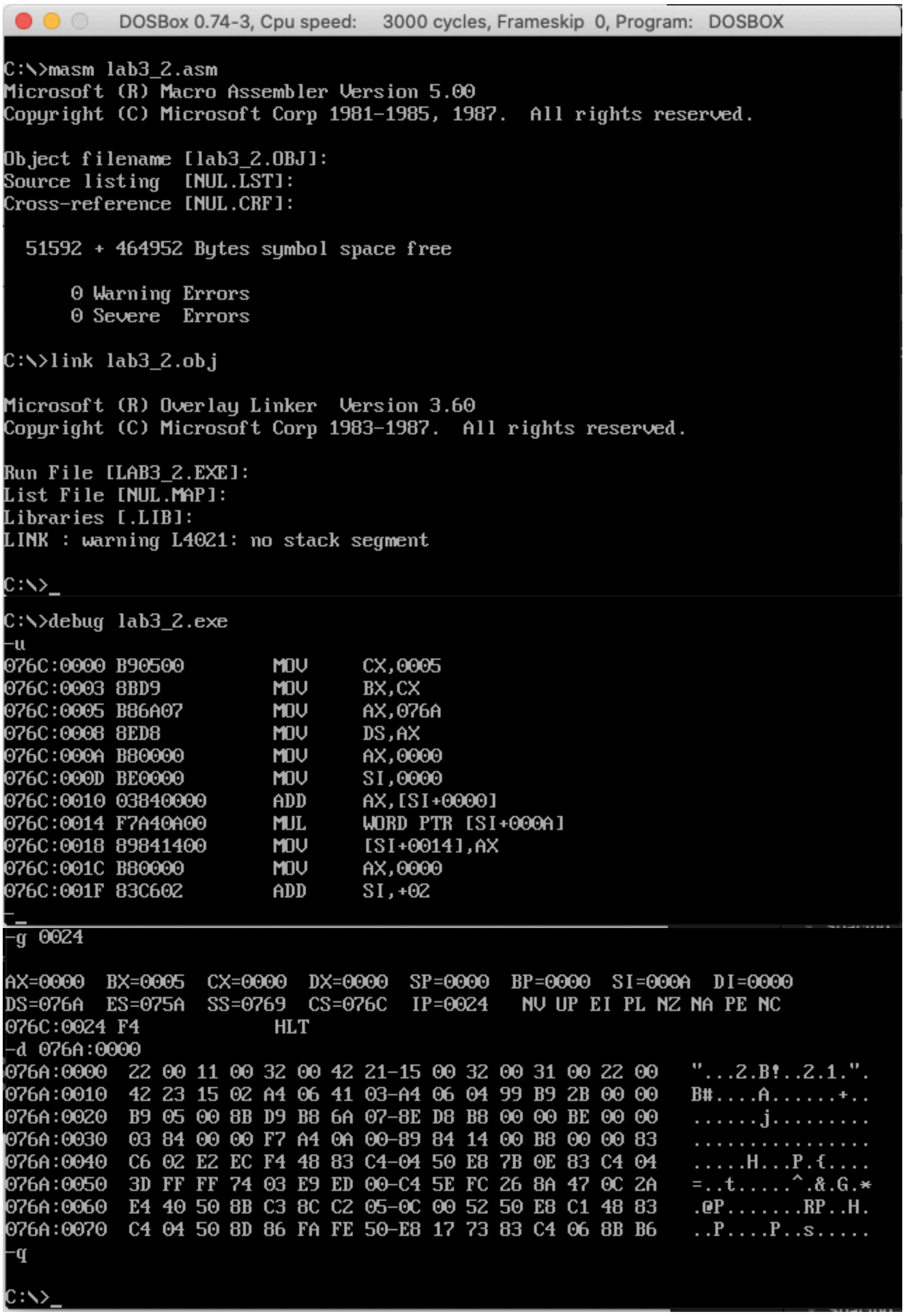
```
C:\>debug lab3_1.exe
-u
076C:0000 B90500        MOV     CX,0005
076C:0003 8BD9          MOV     BX,CX
076C:0005 B86A07        MOV     AX,076A
076C:0008 8ED8          MOV     DS,AX
076C:000A B80000        MOV     AX,0000
076C:000D BE0000        MOV     SI,0000
076C:0010 03840000      ADD     AX,[SI+0000]
076C:0014 03840A00      ADD     AX,[SI+000A]
076C:0018 89841400      MOV     [SI+0014],AX
076C:001C B80000        MOV     AX,0000
076C:001F 83C602        ADD     SI,+02
-_
-u
076C:0022 E2EC          LOOP    0010
076C:0024 F4            HLT
076C:0025 48            DEC     AX
076C:0026 83C404        ADD     SP,+04
076C:0029 50            PUSH    AX
076C:002A E87B0E        CALL    0EA8
076C:002D 83C404        ADD     SP,+04
076C:0030 3DFFFF        CMP     AX,FFFF
076C:0033 7403          JZ      0038
076C:0035 E9ED00        JMP     0125
076C:0038 C45EFC        LES     BX,[BP-04]
076C:003B 26            ES:
076C:003C 8A470C        MOV     AL,[BX+0C]
076C:003F 2AE4          SUB     AH,AH
076C:0041 40            INC     AX
-g 0024

AX=0000  BX=0005  CX=0000  DX=0000  SP=0000  BP=0000  SI=000A  DI=0000
DS=076A  ES=075A  SS=0769  CS=076C  IP=0024   NV UP EI PL NZ NA PE NC
076C:0024 F4            HLT
-
-d 076A:0000
076A:0000  22 00 11 00 32 00 42 21-15 00 32 00 31 00 22 00   "...2.B!..2.1.".
076A:0010  42 23 15 02 54 00 42 00-54 00 84 44 2A 02 00 00   B#..T.B.T..D*...
076A:0020  B9 05 00 8B D9 B8 6A 07-8E D8 B8 00 00 BE 00 00   ......j.........
076A:0030  03 84 00 00 03 84 0A 00-89 84 14 00 B8 00 00 83   ................
076A:0040  C6 02 E2 EC F4 48 83 C4-04 50 E8 7B 0E 83 C4 04   .....H...P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-q

C:\>
```

Task 2

Aim :- Calculating the product of the elements of two arrays.

```
DATA SEGMENT
    MAT1 dw 0022H, 0011H, 0032H, 2142H, 0015H
    MAT2 dw 0032H, 0031H, 0022H, 2342H, 0215H
    RESMAT dw 5 dup(0)
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:
    mov cx, 05H
    mov bx, cx
    mov ax, DATA
    mov ds, ax
    mov ax, 0000H
    mov si, 00H
    RPT:
        add ax, MAT1[si]
        mul MAT2[si]
        mov RESMAT[si], ax
        mov ax, 00H
        add si, 02H
    LOOP RPT
    hlt
CODE ENDS
END START
```

Output


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>masm lab3_2.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab3_2.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51592 + 464952 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab3_2.obj

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

Run File [LAB3_2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>_
```

```
C:\>debug lab3_2.exe
-u
076C:0000 B90500        MOV     CX,0005
076C:0003 8BD9          MOV     BX,CX
076C:0005 B86A07        MOV     AX,076A
076C:0008 8ED8          MOV     DS,AX
076C:000A B80000        MOV     AX,0000
076C:000D BE0000        MOV     SI,0000
076C:0010 03840000      ADD     AX,[SI+0000]
076C:0014 F7A40A00      MUL     WORD PTR [SI+000A]
076C:0018 89841400      MOV     [SI+0014],AX
076C:001C B80000        MOV     AX,0000
076C:001F 83C602        ADD     SI,+02
-_
```

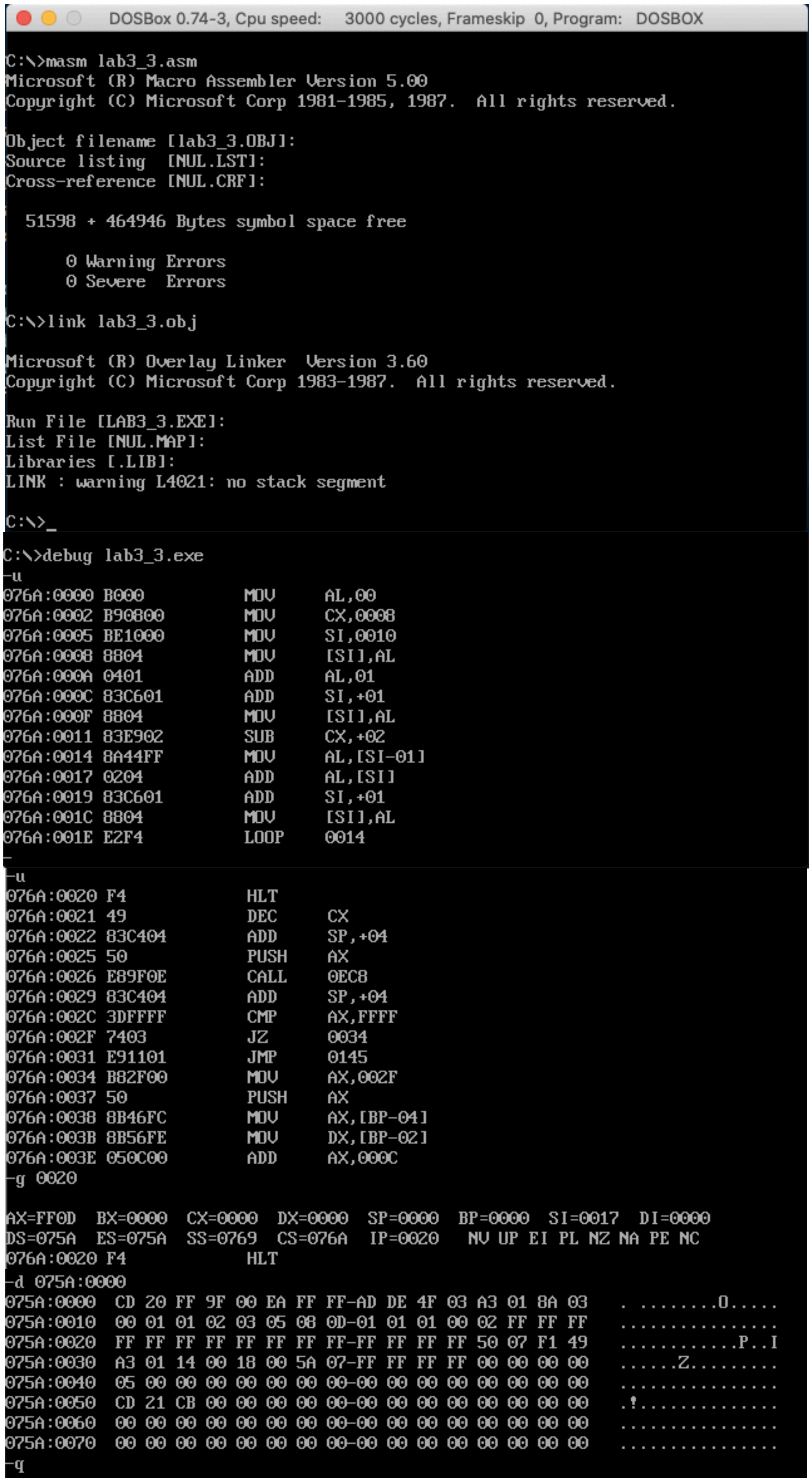
```
-g 0024

AX=0000  BX=0005  CX=0000  DX=0000  SP=0000  BP=0000  SI=000A  DI=0000
DS=076A  ES=075A  SS=0769  CS=076C  IP=0024   NV UP EI PL NZ NA PE NC
076C:0024 F4            HLT
-d 076A:0000
076A:0000  22 00 11 00 32 00 42 21-15 00 32 00 31 00 22 00   "...2.B!..2.1.".
076A:0010  42 23 15 02 A4 06 41 03-A4 06 04 99 B9 2B 00 00   B#....A......+..
076A:0020  B9 05 00 8B D9 B8 6A 07-8E D8 B8 00 00 BE 00 00   ......j.........
076A:0030  03 84 00 00 F7 A4 0A 00-89 84 14 00 B8 00 00 83   ................
076A:0040  C6 02 E2 EC F4 48 83 C4-04 50 E8 7B 0E 83 C4 04   .....H...P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-q

C:\>_
```

Task 3

Aim :- To Generate a Fibonacci Series of numbers.

```
CODE SEGMENT
ASSUME cs:code
START:
    mov al,00H;
    mov cx,08h;
    mov si,10H;
    mov [si],al;
    add al,01h;
    add si,01h;
    mov [si],al;
    sub cx,02h;
    FIBO:
        mov al,[si-1];
        add al,[si];
        add si,01h;
        mov [si],al;
    loop FIBO
    hlt
CODE ENDS
END START
```

Output



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>masm lab3_3.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab3_3.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51598 + 464946 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab3_3.obj

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

Run File [LAB3_3.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>_
```

```
C:\>debug lab3_3.exe
-u
076A:0000 B000          MOV     AL,00
076A:0002 B90800        MOV     CX,0008
076A:0005 BE1000        MOV     SI,0010
076A:0008 8804          MOV     [SI],AL
076A:000A 0401          ADD     AL,01
076A:000C 83C601        ADD     SI,+01
076A:000F 8804          MOV     [SI],AL
076A:0011 83E902        SUB     CX,+02
076A:0014 8A44FF        MOV     AL,[SI-01]
076A:0017 0204          ADD     AL,[SI]
076A:0019 83C601        ADD     SI,+01
076A:001C 8804          MOV     [SI],AL
076A:001E E2F4          LOOP    0014
-
```

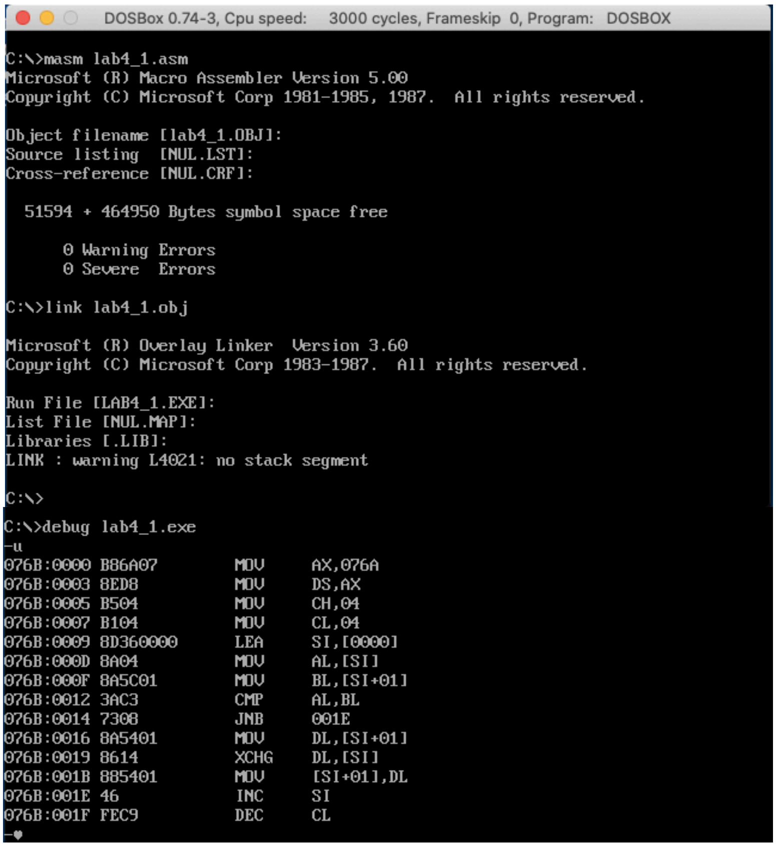
```
-u
076A:0020 F4            HLT
076A:0021 49            DEC     CX
076A:0022 83C404        ADD     SP,+04
076A:0025 50            PUSH    AX
076A:0026 E89F0E        CALL    0EC8
076A:0029 83C404        ADD     SP,+04
076A:002C 3DFFFF        CMP     AX,FFFF
076A:002F 7403          JZ      0034
076A:0031 E91101        JMP     0145
076A:0034 B82F00        MOV     AX,002F
076A:0037 50            PUSH    AX
076A:0038 8B46FC        MOV     AX,[BP-04]
076A:003B 8B56FE        MOV     DX,[BP-02]
076A:003E 050C00        ADD     AX,000C
-g 0020

AX=FF0D  BX=0000  CX=0000  DX=0000  SP=0000  BP=0000  SI=0017  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0020   NV UP EI PL NZ NA PE NC
076A:0020 F4            HLT
-d 075A:0000
075A:0000  CD 20 FF 9F 00 EA FF FF-AD DE 4F 03 A3 01 8A 03   . ........O.....
075A:0010  00 01 01 02 03 05 08 0D-01 01 01 00 02 FF FF FF   ................
075A:0020  FF FF FF FF FF FF FF FF-FF FF FF FF 50 07 F1 49   ............P..I
075A:0030  A3 01 14 00 18 00 5A 07-FF FF FF FF 00 00 00 00   ......Z.........
075A:0040  05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
075A:0050  CD 21 CB 00 00 00 00 00-00 00 00 00 00 00 00 00   .!..............
075A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
075A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-q
```

LAB - 4

Task 1

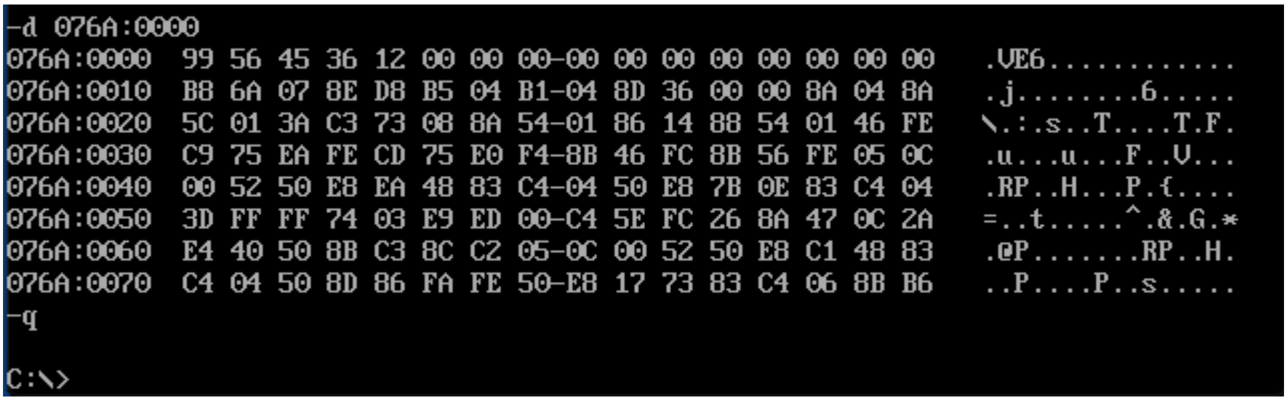
Aim :- Sort an Array in descending order

```
DATA SEGMENT
    STRING1 DB 99H,12H,56H,45H,36H
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:
    MOV AX,DATA
    MOV DS,AX
    MOV CH,04H
    UP2:
        MOV CL,04H
        LEA SI,STRING1
    UP1:
        MOV AL,[SI]
        MOV BL,[SI+1]
        CMP AL,BL
        JNC DOWN
        MOV DL,[SI+1]
        XCHG [SI],DL
        MOV [SI+1],DL

    DOWN:
        INC SI
        DEC CL
        JNZ UP1
        DEC CH
        JNZ UP2
    hlt
CODE ENDS
END START
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>masm lab4_1.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab4_1.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51594 + 464950 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab4_1.obj

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

Run File [LAB4_1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>
C:\>debug lab4_1.exe
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 B504          MOV     CH,04
076B:0007 B104          MOV     CL,04
076B:0009 8D360000      LEA     SI,[0000]
076B:000D 8A04          MOV     AL,[SI]
076B:000F 8A5C01        MOV     BL,[SI+01]
076B:0012 3AC3          CMP     AL,BL
076B:0014 7308          JNB     001E
076B:0016 8A5401        MOV     DL,[SI+01]
076B:0019 8614          XCHG    DL,[SI]
076B:001B 885401        MOV     [SI+01],DL
076B:001E 46            INC     SI
076B:001F FEC9          DEC     CL
-♥
```

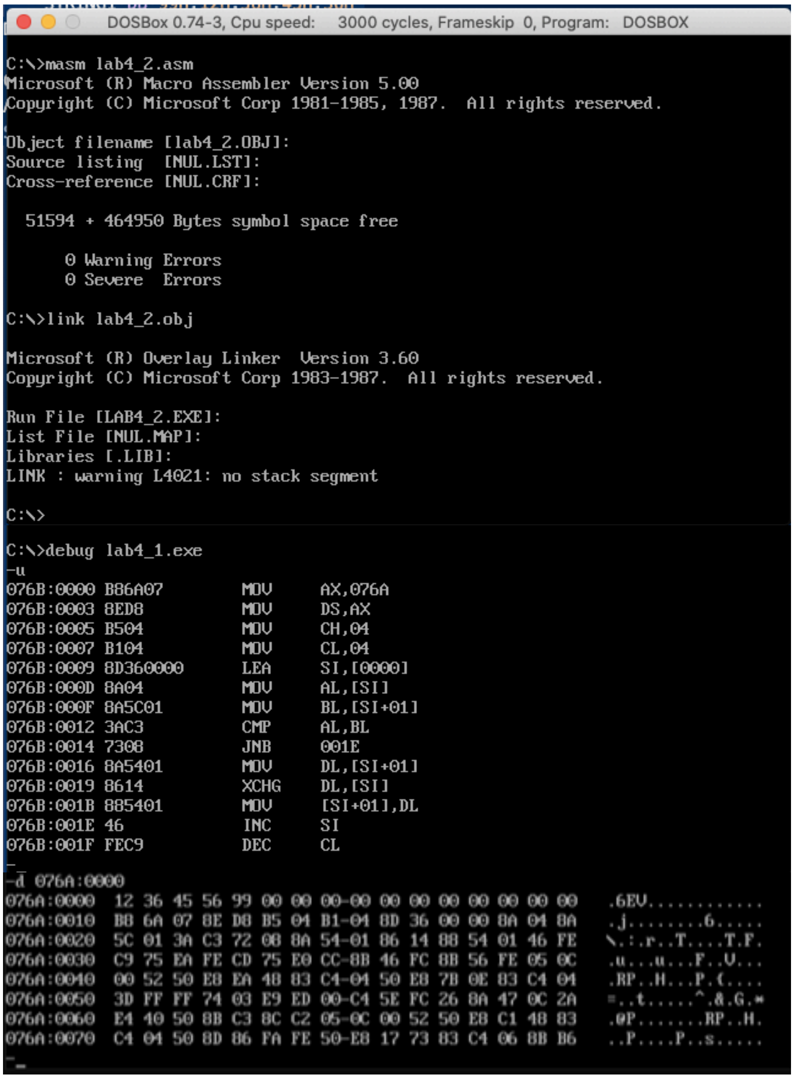


```
-d 076A:0000
076A:0000  99 56 45 36 12 00 00 00-00 00 00 00 00 00 00 00   .VE6............
076A:0010  B8 6A 07 8E D8 B5 04 B1-04 8D 36 00 00 8A 04 8A   .j........6.....
076A:0020  5C 01 3A C3 73 08 8A 54-01 86 14 88 54 01 46 FE   \.:.s..T....T.F.
076A:0030  C9 75 EA FE CD 75 E0 F4-8B 46 FC 8B 56 FE 05 0C   .u...u...F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-q

C:\>
```

Task 2

Aim :- Sort an Array in ascending order

```
DATA SEGMENT
    STRING1 DB 99H,12H,56H,45H,36H
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:
    MOV AX,DATA
    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
    hlt
CODE ENDS
END START
```



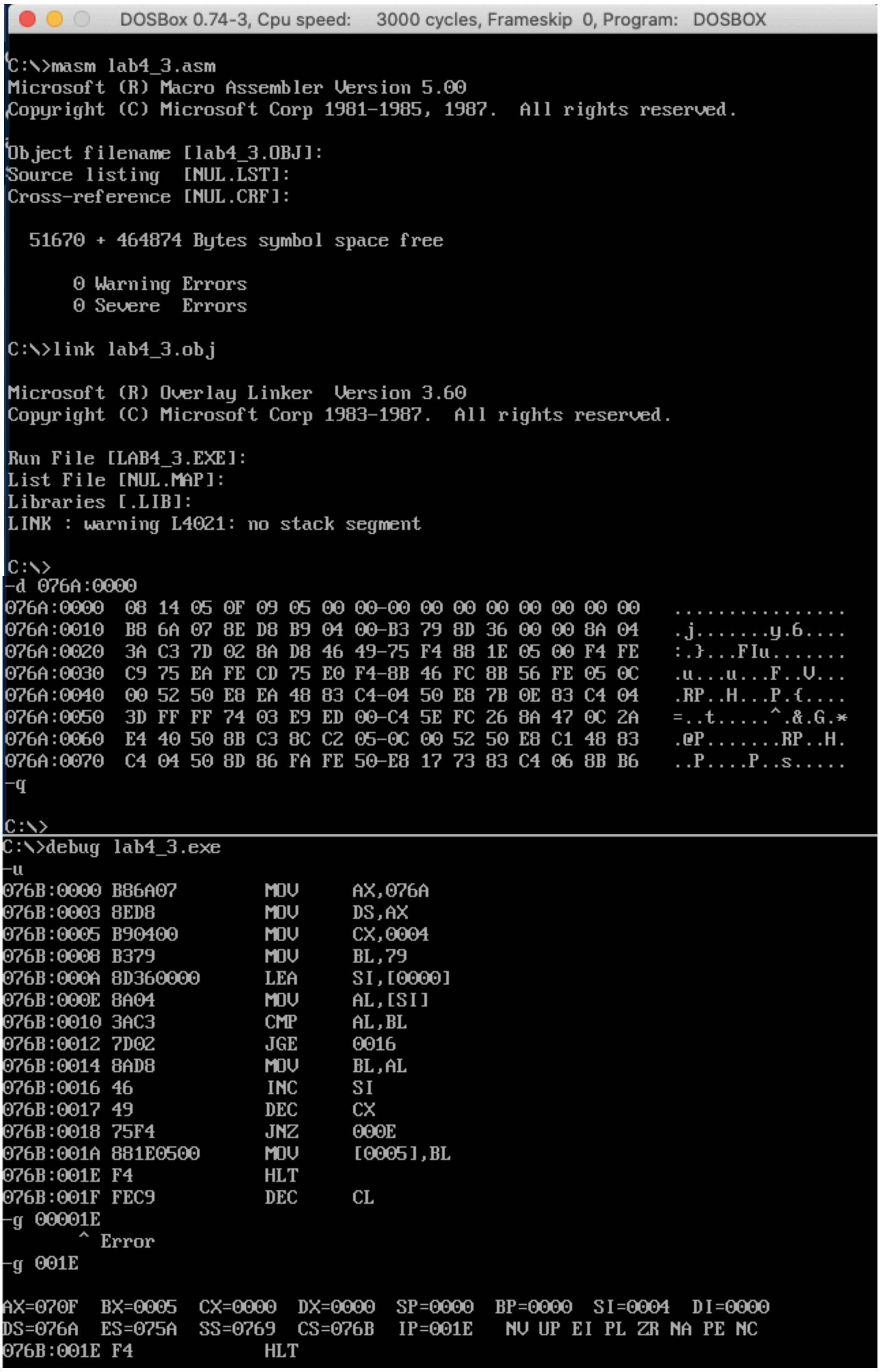
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>masm lab4_2.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab4_2.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51594 + 464950 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab4_2.obj

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

Run File [LAB4_2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>

C:\>debug lab4_1.exe
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 B504          MOV     CH,04
076B:0007 B104          MOV     CL,04
076B:0009 8D360000      LEA     SI,[0000]
076B:000D 8A04          MOV     AL,[SI]
076B:000F 8A5C01        MOV     BL,[SI+01]
076B:0012 3AC3          CMP     AL,BL
076B:0014 7308          JNB     001E
076B:0016 8A5401        MOV     DL,[SI+01]
076B:0019 8614          XCHG    DL,[SI]
076B:001B 885401        MOV     [SI+01],DL
076B:001E 46            INC     SI
076B:001F FEC9          DEC     CL
-
-d 076A:0000
076A:0000  12 36 45 56 99 00 00 00-00 00 00 00 00 00 00 00   .6EV............
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.
076A:0030  C9 75 EA FE CD 75 E0 CC-8B 46 FC 8B 56 FE 05 0C   .u...u...F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-
```

Task 3

Aim :- Find the smallest number in an array

```
data segment
    STRING1 DB 08h,14h,05h,0Fh,09h
    res db ?
data ends

code segment
assume cs:code, ds:data
start:
    mov ax, data
    mov ds, ax
    mov cx, 04h
    mov bl, 79h
    LEA SI, STRING1
    up:
        mov al, [SI]
        cmp al, bl
        jge nxt
        mov bl, al
    nxt:
        inc si
        dec cx
        jnz up
        mov res,bl
    hlt
code ends
end start
```

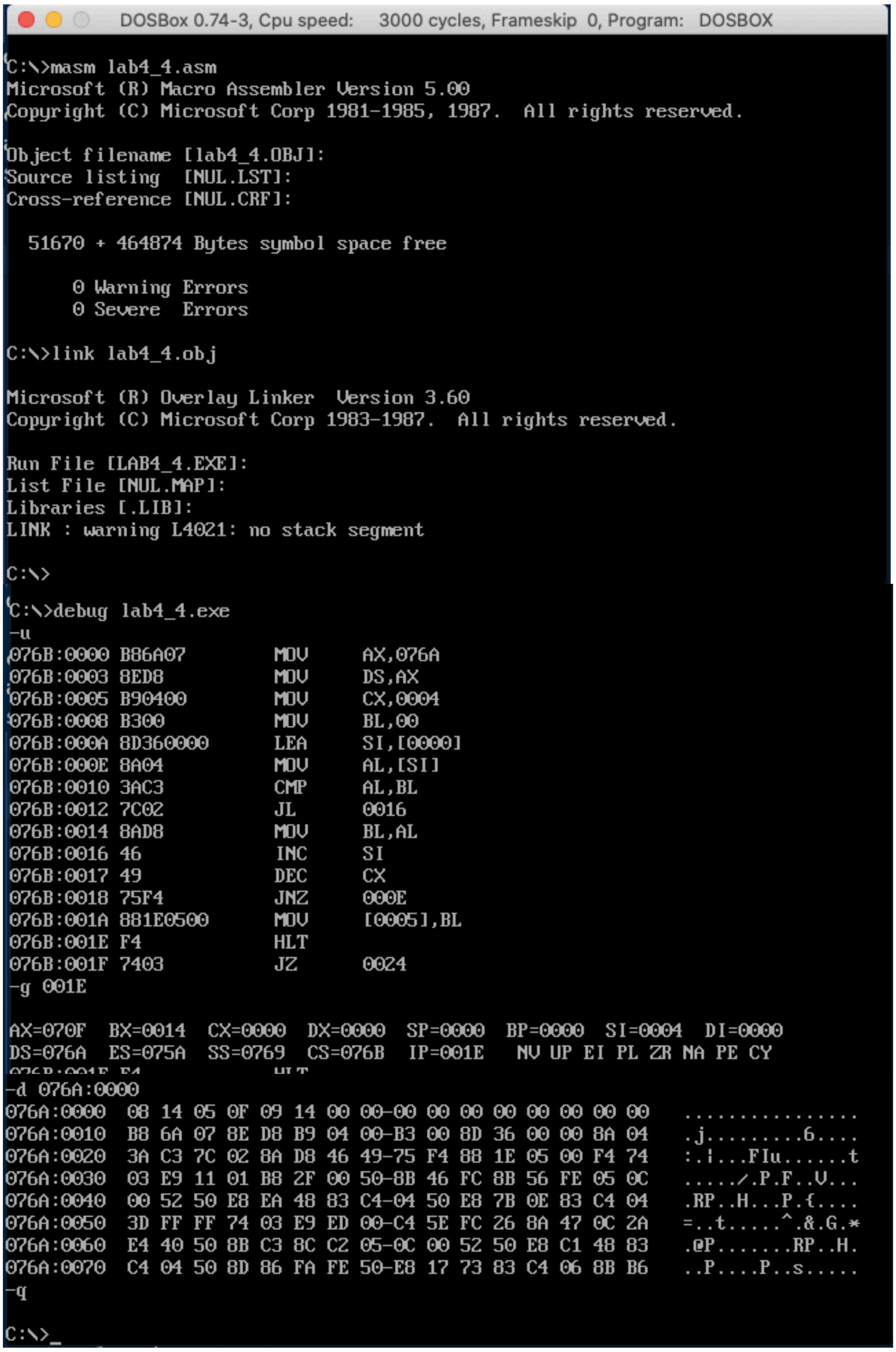

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>masm lab4_3.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab4_3.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51670 + 464874 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab4_3.obj

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

Run File [LAB4_3.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>
-d 076A:0000
076A:0000  08 14 05 0F 09 05 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  B8 6A 07 8E D8 B9 04 00-B3 79 8D 36 00 00 8A 04   .j......y.6....
076A:0020  3A C3 7D 02 8A D8 46 49-75 F4 88 1E 05 00 F4 FE   :.}...FIu.......
076A:0030  C9 75 EA FE CD 75 E0 F4-8B 46 FC 8B 56 FE 05 0C   .u...u...F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-q

C:\>
C:\>debug lab4_3.exe
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 B90400        MOV     CX,0004
076B:0008 B379          MOV     BL,79
076B:000A 8D360000      LEA     SI,[0000]
076B:000E 8A04          MOV     AL,[SI]
076B:0010 3AC3          CMP     AL,BL
076B:0012 7D02          JGE     0016
076B:0014 8AD8          MOV     BL,AL
076B:0016 46            INC     SI
076B:0017 49            DEC     CX
076B:0018 75F4          JNZ     000E
076B:001A 881E0500      MOV     [0005],BL
076B:001E F4            HLT
076B:001F FEC9          DEC     CL
-g 00001E
      ^ Error
-g 001E

AX=070F  BX=0005  CX=0000  DX=0000  SP=0000  BP=0000  SI=0004  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=001E   NV UP EI PL ZR NA PE NC
076B:001E F4            HLT
```

Task 4

Aim :- Find the largest number in an array

```
data segment
    STRING1 DB 08h,14h,05h,0Fh,09h
    res db ?
data ends

code segment
assume cs:code, ds:data
start:
    mov ax, data
    mov ds, ax
    mov cx, 04h
    mov bl, 00h
    LEA SI, STRING1
    up:
        mov al, [SI]
        cmp al, bl
        jl nxt
        mov bl, al
    nxt:
        inc si
        dec cx
        jnz up
        mov res,bl
    hlt
code ends
end start
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>masm lab4_4.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab4_4.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51670 + 464874 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab4_4.obj

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

Run File [LAB4_4.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>

C:\>debug lab4_4.exe
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 B90400        MOV     CX,0004
076B:0008 B300          MOV     BL,00
076B:000A 8D360000      LEA     SI,[0000]
076B:000E 8A04          MOV     AL,[SI]
076B:0010 3AC3          CMP     AL,BL
076B:0012 7C02          JL      0016
076B:0014 8AD8          MOV     BL,AL
076B:0016 46            INC     SI
076B:0017 49            DEC     CX
076B:0018 75F4          JNZ     000E
076B:001A 881E0500      MOV     [0005],BL
076B:001E F4            HLT
076B:001F 7403          JZ      0024
-g 001E

AX=070F  BX=0014  CX=0000  DX=0000  SP=0000  BP=0000  SI=0004  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=001E   NV UP EI PL ZR NA PE CY
076B:001E F4            HLT
-d 076A:0000
076A:0000  08 14 05 0F 09 14 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  B8 6A 07 8E D8 B9 04 00-B3 00 8D 36 00 00 8A 04   .j.........6....
076A:0020  3A C3 7C 02 8A D8 46 49-75 F4 88 1E 05 00 F4 74   :.|...FIu......t
076A:0030  03 E9 11 01 B8 2F 00 50-8B 46 FC 8B 56 FE 05 0C   ...../.P.F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-q

C:\>_
```

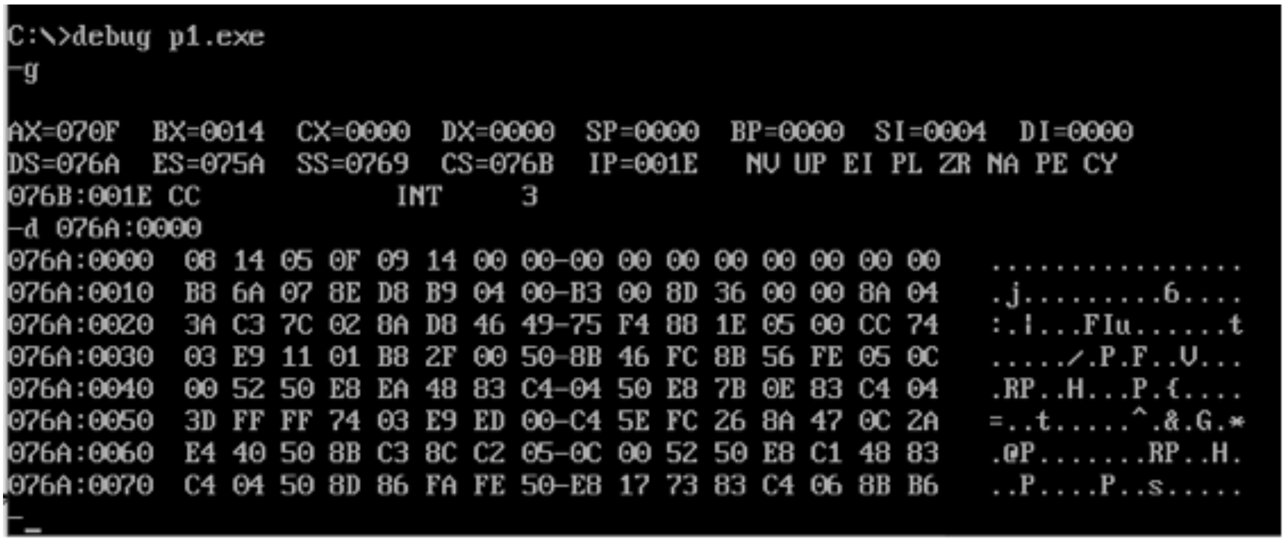
)

LAB - 5

Task 1

Aim :- Convert a BCD number into a Hexadecimal number

```
data segment
    bcd dw 25h
    bin dw ?
data ends
code segment
assume cs:code,ds:data
start:
    mov ax,data
    mov ds,ax
    mov ax,bcd
    mov ax,0fh
    mov bx,ax
    mov ax,bcd
    ans ax,0f0h
    mov cl,04h
    ror al,cl
    mov cx,0ah
    mul cx
    add ax,bx
    mov bin,ax
    mov ah,4ch
    hlt
code ends
end start
```



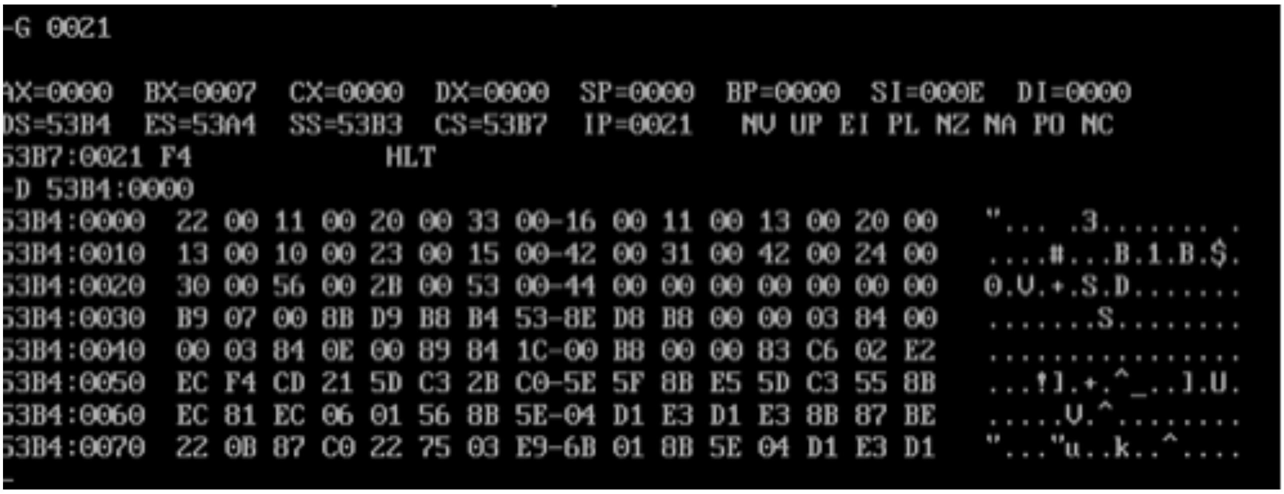
```
C:\>debug p1.exe
-g

AX=070F  BX=0014  CX=0000  DX=0000  SP=0000  BP=0000  SI=0004  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=001E   NV UP EI PL ZR NA PE CY
076B:001E CC            INT     3
-d 076A:0000
076A:0000  08 14 05 0F 09 14 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  B8 6A 07 8E D8 B9 04 00-B3 00 8D 36 00 00 8A 04   .j.........6....
076A:0020  3A C3 7C 02 8A D8 46 49-75 F4 88 1E 05 00 CC 74   :.|...FIu......t
076A:0030  03 E9 11 01 B8 2F 00 50-8B 46 FC 8B 56 FE 05 0C   ...../.P.F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-_
```

Task 2

Aim :- Find the sum of two matrices

```
assume cs:code, ds:data
data segment
    mat1 dw 0022h, 0011h, 0020h, 0033h, 0016h, 0011h, 
0013h
    mat2 dw 0020h, 0013h, 0010h, 0023h, 0015h, 0042h, 
0031h
    resmat dw 7 dup(0)
data ends

code segment
start:
    mov cx, 07h
    mov bx, cx
    mov ax, data
    mov ds, ax
    mov ax, 00h
```

```
rpt:
    add ax, mat1[si]
    add ax, mat2[si]
    mov resmat[si], ax
    mov ax, 00h
    add si, 02h
    loop rpt
    hlt
    int 21h
code ends
end start
```



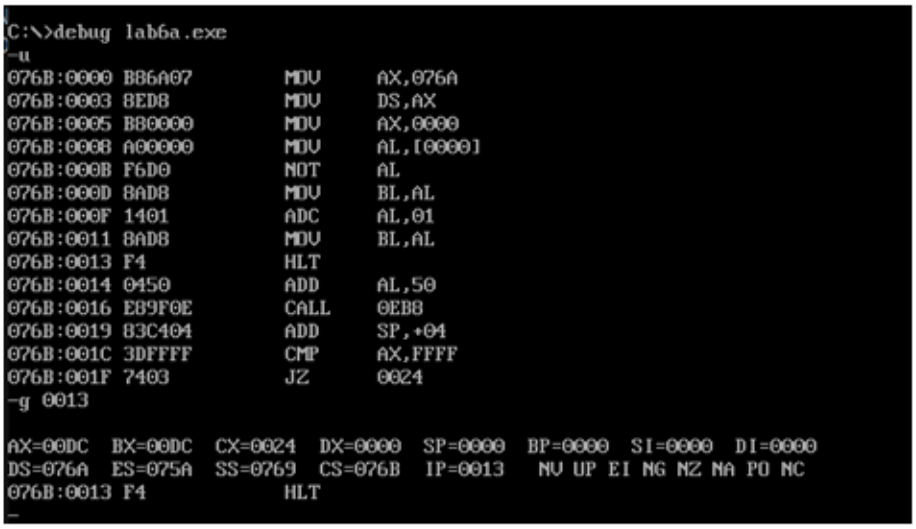
```
-G 0021

AX=0000  BX=0007  CX=0000  DX=0000  SP=0000  BP=0000  SI=000E  DI=0000
DS=53B4  ES=53A4  SS=53B3  CS=53B7  IP=0021   NV UP EI PL NZ NA PO NC
53B7:0021 F4            HLT
-D 53B4:0000
53B4:0000  22 00 11 00 20 00 33 00-16 00 11 00 13 00 20 00   "... .3....... .
53B4:0010  13 00 10 00 23 00 15 00-42 00 31 00 42 00 24 00   ....#...B.1.B.$.
53B4:0020  30 00 56 00 2B 00 53 00-44 00 00 00 00 00 00 00   0.V.+.S.D.......
53B4:0030  B9 07 00 8B D9 B8 B4 53-8E D8 B8 00 00 03 84 00   .......S........
53B4:0040  00 03 84 0E 00 89 84 1C-00 B8 00 00 83 C6 02 E2   ................
53B4:0050  EC F4 CD 21 5D C3 2B C0-5E 5F 8B E5 5D C3 55 8B   ...!].+.^_..].U.
53B4:0060  EC 81 EC 06 01 56 8B 5E-04 D1 E3 D1 E3 8B 87 BE   .....V.^........
53B4:0070  22 0B 87 C0 22 75 03 E9-6B 01 8B 5E 04 D1 E3 D1   "..."u..k..^....
-
```

LAB - 6

Task 1

Aim :- To find the 2's complement of a number

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
    VAR1 DB 24H
DATA ENDS
CODE SEGMENT
START:
    MOV AX, DATA
    MOV DS,AX
    MOV AX,0000H
    MOV AL,VAR1
    NOT AL
    MOV BL,AL
    ADC AL,00000001B
    MOV BL,AL
    HLT

CODE ENDS
END START
```

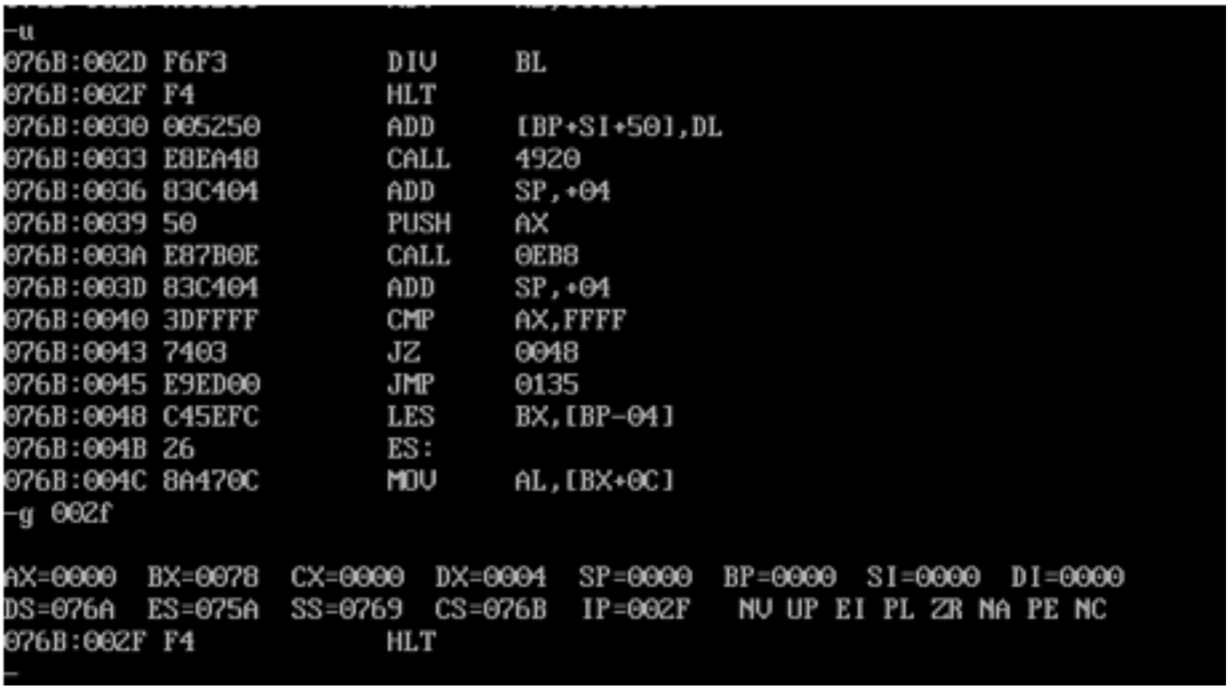


```
C:\>debug lab6a.exe
-u
076B:0000 BB6A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 B80000        MOV     AX,0000
076B:0008 A00000        MOV     AL,[0000]
076B:000B F6D0          NOT     AL
076B:000D 8AD8          MOV     BL,AL
076B:000F 1401          ADC     AL,01
076B:0011 8AD8          MOV     BL,AL
076B:0013 F4            HLT
076B:0014 0450          ADD     AL,50
076B:0016 E89F0E        CALL    0EB8
076B:0019 83C404        ADD     SP,+04
076B:001C 3DFFFF        CMP     AX,FFFF
076B:001F 7403          JZ      0024
-g 0013

AX=00DC  BX=00DC  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 F4            HLT
-
```

Task 2

Aim :- To find the permutation n!/(n-r)! Given values of n and r

```
ASSUME CS:CODE, DS:DATA

DATA SEGMENT
    NUM DB 05H
    RE DB 02H
    NUM1 DB 00H
    DEN DB 00H
DATA ENDS

CODE SEGMENT
FACT PROC NEAR
    CMP CL,00H
    JE RETURN
    MUL CL
    DEC CL
    CALL FACT
RETURN: RET
FACT ENDP

START:
    MOV AX,DATA
    MOV DS,AX
    MOV AX,0000H
    MOV AL,NUM
    MOV DL,NUM
    SUB DL,01H
    MOV BL,RE
    MOV CL,DL
    CALL FACT
    MOV BL,AL
    MOV AL,NUM1
    DIV BL
    HLT
CODE ENDS
END START
```

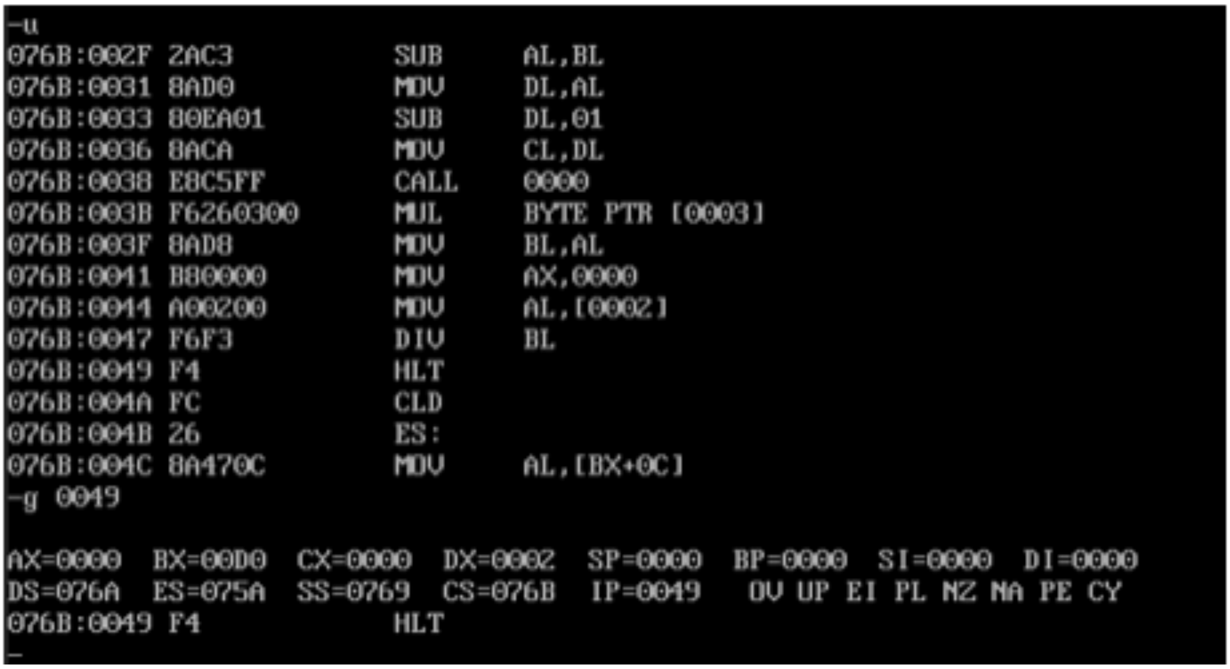


```
-u
076B:002D F6F3          DIV     BL
076B:002F F4            HLT
076B:0030 005250        ADD     [BP+SI+50],DL
076B:0033 E8EA48        CALL    4920
076B:0036 83C404        ADD     SP,+04
076B:0039 50            PUSH    AX
076B:003A E87B0E        CALL    0EB8
076B:003D 83C404        ADD     SP,+04
076B:0040 3DFFFF        CMP     AX,FFFF
076B:0043 7403          JZ      0048
076B:0045 E9ED00        JMP     0135
076B:0048 C45EFC        LES     BX,[BP-04]
076B:004B 26            ES:
076B:004C 8A470C        MOV     AL,[BX+0C]
-g 002f

AX=0000  BX=0078  CX=0000  DX=0004  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=002F   NV UP EI PL ZR NA PE NC
076B:002F F4            HLT
-
```

Task 3

Aim :- To find the combination n!/r!*(n-r)! Given values of n and r

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
    NUM DB 05H
    RE DB 02H
    NUM1 DB 00H
    DEN DB 00H
DATA ENDS
CODE SEGMENT

FACT PROC NEAR
    CMP CL,00H
    JE RETURN
    MUL CL
    DEC CL
    CALL FACT
    RETURN: RET
FACT ENDP
START:
    MOV AX,DATA
    MOV DS,AX
    MOV AX,0000H
    MOV AL,NUM
    MOV DL,AL
    SUB DL,01H
    MOV CL,DL
    CALL FACT
    MOV DEN,AL
    MOV AX,0000H
    MOV AL,NUM
    MOV BL,RE
    SUB AL,BL
    MOV DL,AL
    SUB DL,01H
    MOV CL,DL
    CALL FACT
    MUL DEN
    MOV BL,AL
    MOV AX, 00H
    MOV AL,NUM1
    DIV BL
    HLT
CODE ENDS
END START
```



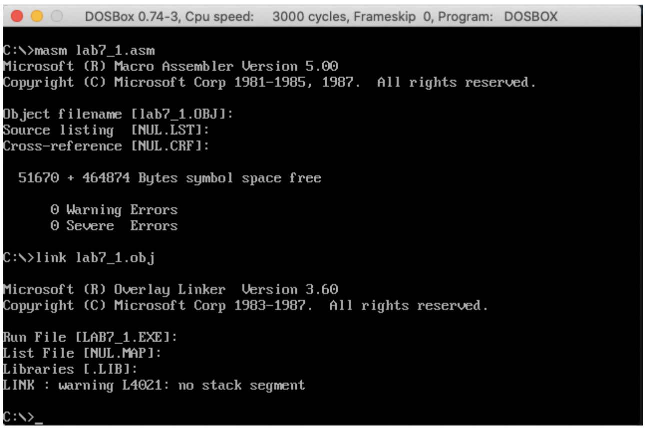
```
-u
076B:002F 2AC3          SUB     AL,BL
076B:0031 8AD0          MOV     DL,AL
076B:0033 80EA01        SUB     DL,01
076B:0036 8ACA          MOV     CL,DL
076B:0038 E8C5FF        CALL    0000
076B:003B F6260300      MUL     BYTE PTR [0003]
076B:003F 8AD8          MOV     BL,AL
076B:0041 B80000        MOV     AX,0000
076B:0044 A00200        MOV     AL,[0002]
076B:0047 F6F3          DIV     BL
076B:0049 F4            HLT
076B:004A FC            CLD
076B:004B 26            ES:
076B:004C 8A470C        MOV     AL,[BX+0C]
-g 0049

AX=0000  BX=00D0  CX=0000  DX=0002  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0049   OV UP EI PL NZ NA PE CY
076B:0049 F4            HLT
-
```

LAB - 7

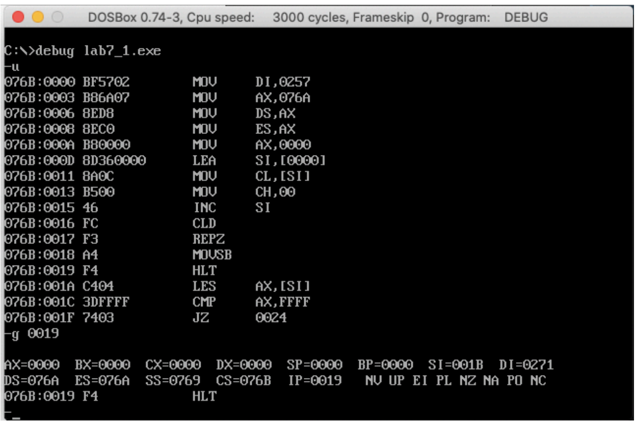
Task 1

Aim :- To Scan a byte from a list of bytes

```
assume cs:code, ds:data
data segment
    r1 db 1ah,2bh,3ch,4dh,5eh,6fh
data ends
code segment
start:
    mov di,599
    mov ax,data
    mov ds,ax
    mov es,ax
    mov ax,0000
    lea si,var1
    mov cl,[si]
    mov ch,00h
    inc si
    cld
    rep movsb
    hlt
code ends
end start
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>masm lab7_1.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab7_1.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51670 + 464874 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link lab7_1.obj

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

Run File [LAB7_1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>_
```

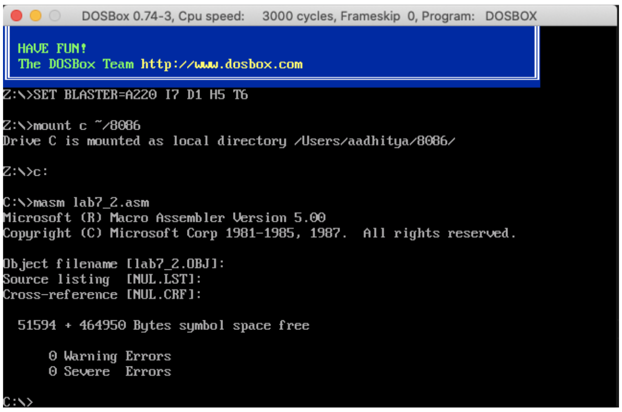

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
C:\>debug lab7_1.exe
-u
076B:0000 BF5702        MOV     DI,0257
076B:0003 B86A07        MOV     AX,076A
076B:0006 8ED8          MOV     DS,AX
076B:0008 8EC0          MOV     ES,AX
076B:000A B80000        MOV     AX,0000
076B:000D 8D360000      LEA     SI,[0000]
076B:0011 8A0C          MOV     CL,[SI]
076B:0013 B500          MOV     CH,00
076B:0015 46            INC     SI
076B:0016 FC            CLD
076B:0017 F3            REPZ
076B:0018 A4            MOVSB
076B:0019 F4            HLT
076B:001A C404          LES     AX,[SI]
076B:001C 3DFFFF        CMP     AX,FFFF
076B:001F 7403          JZ      0024
-g 0019

AX=0000  BX=0000  CX=0000  DX=0000  SP=0000  BP=0000  SI=001B  DI=0271
DS=076A  ES=076A  SS=0769  CS=076B  IP=0019   NV UP EI PL NZ NA PO NC
076B:0019 F4            HLT
-_
```

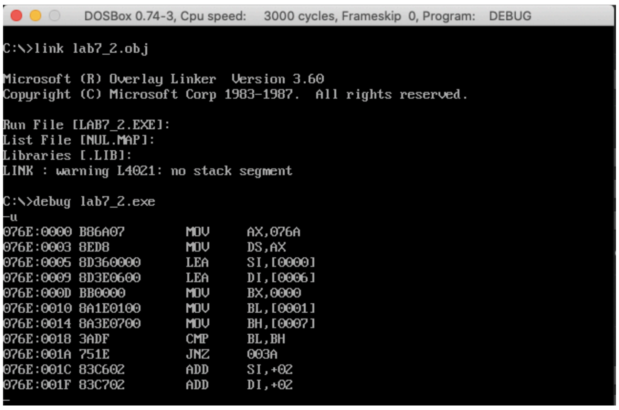
Task 2

Aim :- To compare two different strings of same length and check if they are equal.

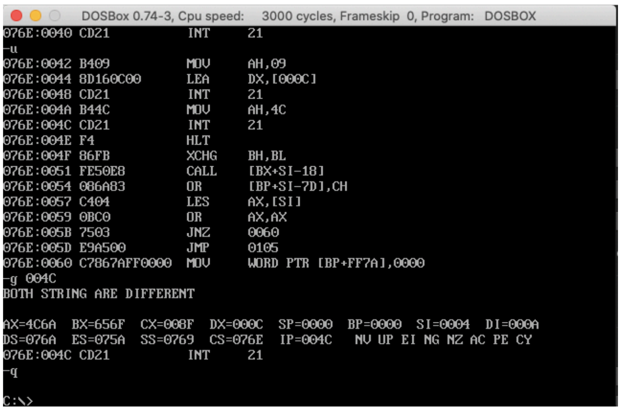
```
DATA SEGMENT
    STR1 DB "hello$"
    STR2 DB "hello$"
    NEWLINE DB 10,13,"$"
    MSG1 DB "BOTH STRING ARE SAME$"
    MSG2 DB "BOTH STRING ARE DIFFERENT$"
DATA ENDS
CODE SEGMENT
    ASSUME DS:DATA,CS:CODE
START:
    MOV AX,DATA
    MOV DS,AX
    LEA SI,STR1
    LEA DI,STR2
STRING_COMPARISION :
    MOV BX,00
    MOV BL,STR1+1
    MOV BH,STR2+1
    CMP BL,BH
    JNE L1
    ADD SI,2
    ADD DI,2
  L2:MOV BL,BYTE PTR[SI]
    CMP BYTE PTR[DI],BL
    JNE L1
    INC SI
    INC DI
    CMP BYTE PTR[DI],"$"
    JNE L2
    MOV AH,09H
    LEA DX,MSG1
    INT 21H
    JMP L5
  L1:MOV AH,09H
    LEA DX,MSG2
    INT 21H
 L5:
    MOV AH,09H
    LEA DX,NEWLINE
    INT 21H
    MOV AH,4CH
    INT 21H
hlt
CODE ENDS
END START
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

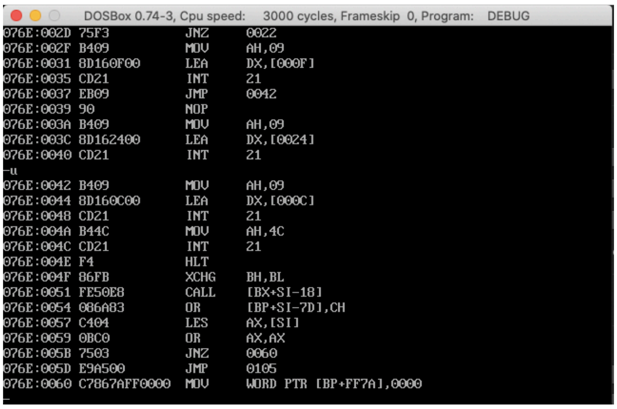
```
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c ~/8086
Drive C is mounted as local directory /Users/aadhitya/8086/

Z:\>c:

C:\>masm lab7_2.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab7_2.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51594 + 464950 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
C:\>link lab7_2.obj

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

Run File [LAB7_2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>debug lab7_2.exe
-u
076E:0000 B86A07        MOV     AX,076A
076E:0003 8ED8          MOV     DS,AX
076E:0005 8D360000      LEA     SI,[0000]
076E:0009 8D3E0600      LEA     DI,[0006]
076E:000D BB0000        MOV     BX,0000
076E:0010 8A1E0100      MOV     BL,[0001]
076E:0014 8A3E0700      MOV     BH,[0007]
076E:0018 3ADF          CMP     BL,BH
076E:001A 751E          JNZ     003A
076E:001C 83C602        ADD     SI,+02
076E:001F 83C702        ADD     DI,+02
-
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
076E:0040 CD21          INT     21
-u
076E:0042 B409          MOV     AH,09
076E:0044 8D160C00      LEA     DX,[000C]
076E:0048 CD21          INT     21
076E:004A B44C          MOV     AH,4C
076E:004C CD21          INT     21
076E:004E F4            HLT
076E:004F 86FB          XCHG    BH,BL
076E:0051 FE50E8        CALL    [BX+SI-18]
076E:0054 086A83        OR      [BP+SI-7D],CH
076E:0057 C404          LES     AX,[SI]
076E:0059 0BC0          OR      AX,AX
076E:005B 7503          JNZ     0060
076E:005D E9A500        JMP     0105
076E:0060 C7867AFF0000  MOV     WORD PTR [BP+FF7A],0000
-g 004C
BOTH STRING ARE DIFFERENT

AX=4C6A  BX=656F  CX=008F  DX=000C  SP=0000  BP=0000  SI=0004  DI=000A
DS=076A  ES=075A  SS=0769  CS=076E  IP=004C   NV UP EI NG NZ AC PE CY
076E:004C CD21          INT     21
-q

C:\>
```

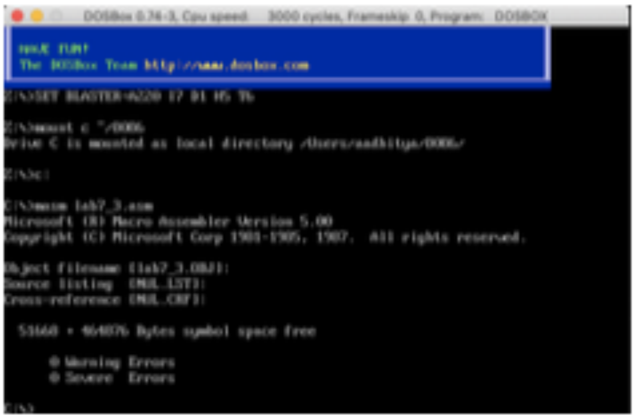

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
076E:002D 75F3          JNZ     0022
076E:002F B409          MOV     AH,09
076E:0031 8D160F00      LEA     DX,[000F]
076E:0035 CD21          INT     21
076E:0037 EB09          JMP     0042
076E:0039 90            NOP
076E:003A B409          MOV     AH,09
076E:003C 8D162400      LEA     DX,[0024]
076E:0040 CD21          INT     21
-u
076E:0042 B409          MOV     AH,09
076E:0044 8D160C00      LEA     DX,[000C]
076E:0048 CD21          INT     21
076E:004A B44C          MOV     AH,4C
076E:004C CD21          INT     21
076E:004E F4            HLT
076E:004F 86FB          XCHG    BH,BL
076E:0051 FE50E8        CALL    [BX+SI-18]
076E:0054 086A83        OR      [BP+SI-7D],CH
076E:0057 C404          LES     AX,[SI]
076E:0059 0BC0          OR      AX,AX
076E:005B 7503          JNZ     0060
076E:005D E9A500        JMP     0105
076E:0060 C7867AFF0000  MOV     WORD PTR [BP+FF7A],0000
-
```

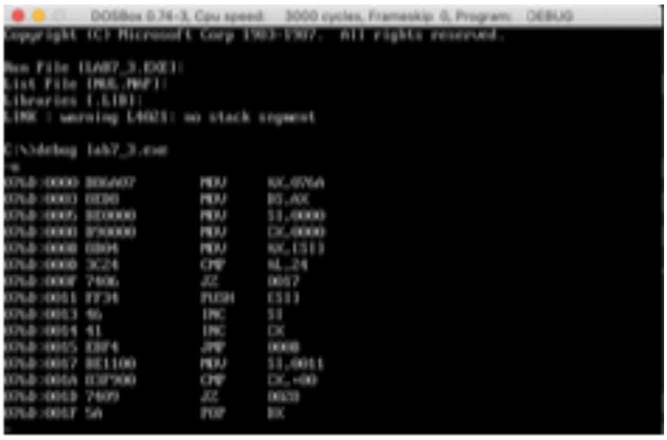
Task 3

Aim :- To Reverse the given string.

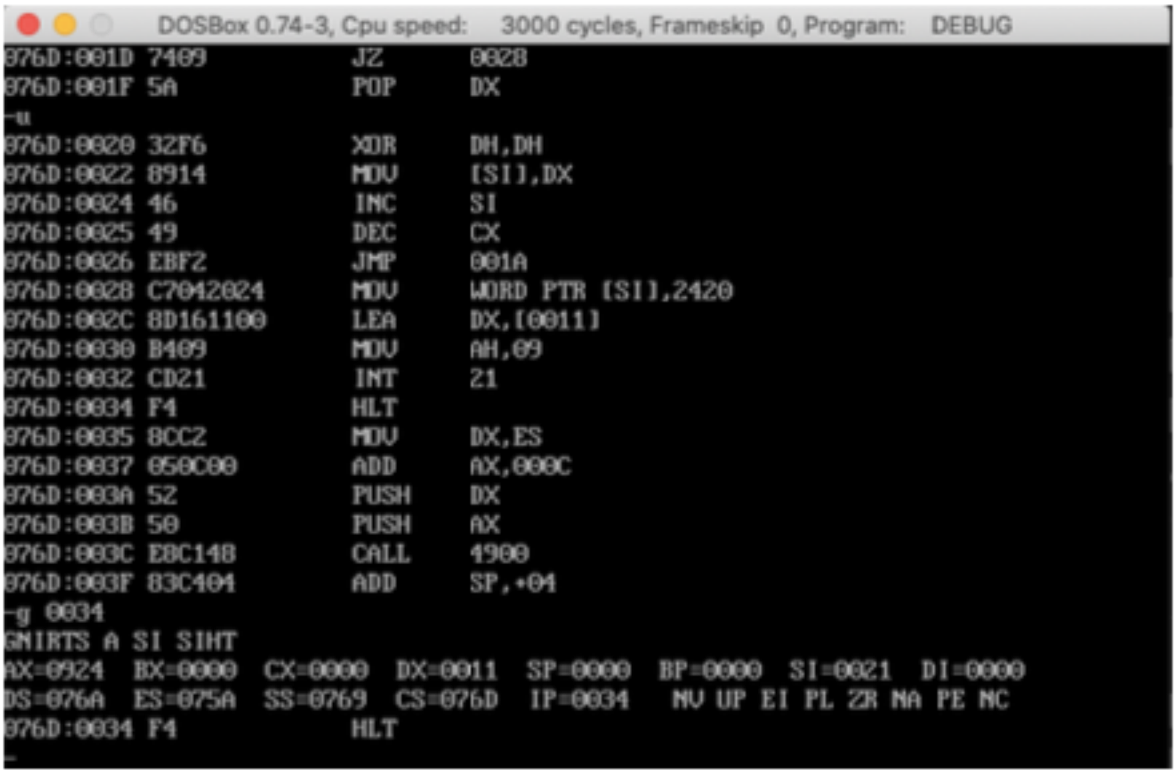
```
DATA SEGMENT
    STR DB "THIS IS A STRING$"
    STR2 DB 30 DUP(0)
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
    START:
        MOV AX,DATA
        MOV DS,AX
        MOV SI,OFFSET STR
        MOV CX,0H

        LOOP1:
            MOV AX,[SI]
            CMP AL,'$'
            JE LABEL1

            PUSH [SI]

            INC SI
            INC CX
            JMP LOOP1

        LABEL1:
            MOV SI, OFFSET STR2

            LOOP2:
                CMP CX,0
                JE EXIT

                POP DX
                XOR DH,DH
                MOV [SI],DX
                INC SI
                DEC CX
                JMP LOOP2

        EXIT:
            MOV [SI],'$ '
            LEA DX, STR2
            MOV AH,09H
            INT 21H
    HLT
CODE ENDS

END START
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c "/8086
Drive C is mounted as local directory /Users/aadhitya/8086/

Z:\>c:

C:\>masm lab7_3.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [lab7_3.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51660 + 464876 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [LAB7_3.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>debug lab7_3.exe
-u
076D:0000 B86A07        MOV     AX,076A
076D:0003 8ED8          MOV     DS,AX
076D:0005 BE0000        MOV     SI,0000
076D:0008 B90000        MOV     CX,0000
076D:000B 8B04          MOV     AX,[SI]
076D:000D 3C24          CMP     AL,24
076D:000F 7406          JZ      0017
076D:0011 FF34          PUSH    [SI]
076D:0013 46            INC     SI
076D:0014 41            INC     CX
076D:0015 EBF4          JMP     000B
076D:0017 BE1100        MOV     SI,0011
076D:001A 83F900        CMP     CX,+00
076D:001D 7409          JZ      0028
076D:001F 5A            POP     DX
-
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
076D:001D 7409          JZ      0028
076D:001F 5A            POP     DX
-u
076D:0020 32F6          XOR     DH,DH
076D:0022 8914          MOV     [SI],DX
076D:0024 46            INC     SI
076D:0025 49            DEC     CX
076D:0026 EBF2          JMP     001A
076D:0028 C70420 24     MOV     WORD PTR [SI],2420
076D:002C 8D161100      LEA     DX,[0011]
076D:0030 B409          MOV     AH,09
076D:0032 CD21          INT     21
076D:0034 F4            HLT
076D:0035 8CC2          MOV     DX,ES
076D:0037 050C00        ADD     AX,000C
076D:003A 52            PUSH    DX
076D:003B 50            PUSH    AX
076D:003C E8C148        CALL    4900
076D:003F 83C404        ADD     SP,+04
-g 0034
GNIRTS A SI SIHT
AX=0924  BX=0000  CX=0000  DX=0011  SP=0000  BP=0000  SI=0021  DI=0000
DS=076A  ES=075A  SS=0769  CS=076D  IP=0034   NV UP EI PL ZR NA PE NC
076D:0034 F4            HLT
-
```

LAB - 8

Task 1

Aim :- To Rotate stepper motor in a) clockwise and b)anticlockwise direction

```
(STARTING ADDRESS = 1000)
MOV CX,04H        (1000)
MOV SI,4200       (1004)
MOV AL,[SI] (1008)
OUT 0C0H,AL       (100A)
MOV BX,0FFFFH     (100C)
DEC BX            (1010)
JNZ 1010     (1011)
INC SI(1013)
LOOP 1008    (1014)
JMP 1000     (1016)
HLT    (1019)
```

a) For clockwise direction:

SB 4200 → 09,05,06,0A

b) For anticlockwise direction

SB 4200 → 0A,06,05,09

LAB - 9

Task 1

Aim :- To convert analog signal into a digital signal

(STARTING ADDRESS = 1000)

```
MOV AL,10H        (1000)
OUT C8H,AL        (1003)
MOV AL,18H         (1005)
OUT C8H,AL        (1008)
HLT               (1009)
```

Code is executed to convert analog signal to digital signal.