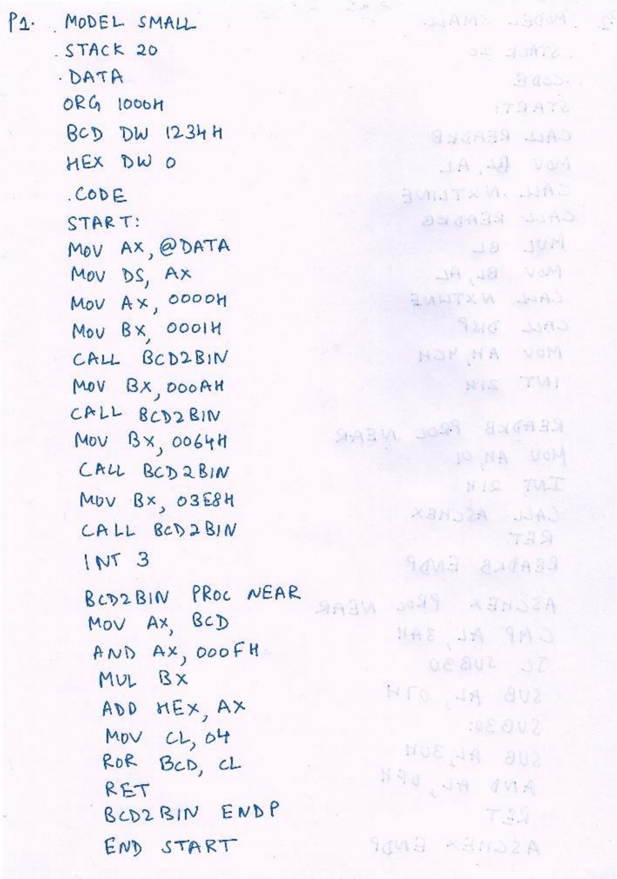
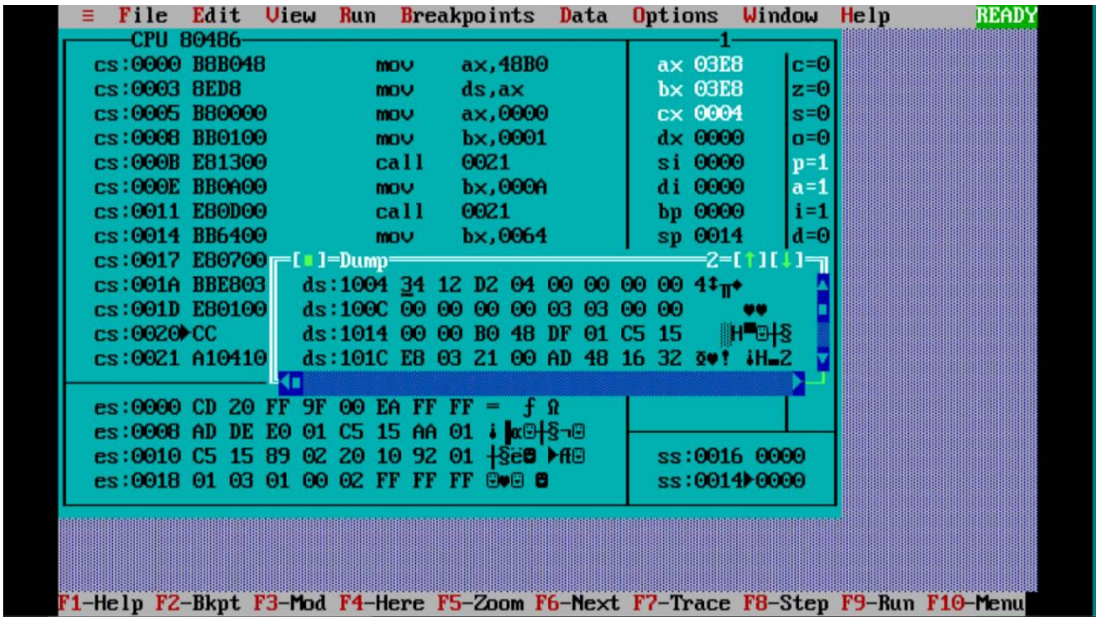
Lab 4

P1: Write a program to convert 4-digit BCD number to HEXADECIMAL number and store the result in memory.



```
P1. .MODEL SMALL
    .STACK 20
    .DATA
    ORG 1000H
    BCD DW 1234H
    HEX DW 0
    .CODE
    START:
    MOV AX, @DATA
    MOV DS, AX
    MOV AX, 0000H
    MOV BX, 0001H
    CALL BCD2BIN
    MOV BX, 000AH
    CALL BCD2BIN
    MOV BX, 0064H
    CALL BCD2BIN
    MOV BX, 03E8H
    CALL BCD2BIN
    INT 3
    BCD2BIN PROC NEAR
    MOV AX, BCD
    AND AX, 000FH
    MUL BX
    ADD HEX, AX
    MOV CL, 04
    ROR BCD, CL
    RET
    BCD2BIN ENDP
    END START
```



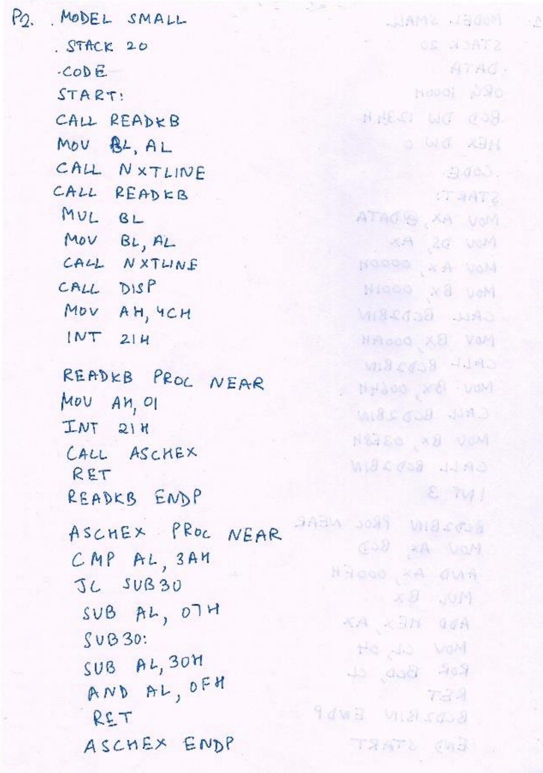
≡ File Edit View Run Breakpoints Data Options Window Help READY
CPU 80486
1
[■]=Dump
2=[↑][↓]
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```
cs:0000 B8B048      mov   ax,48B0        ax 03E8   c=0
cs:0003 8ED8        mov   ds,ax          bx 03E8   z=0
cs:0005 B80000      mov   ax,0000        cx 0004   s=0
cs:0008 BB0100      mov   bx,0001        dx 0000   o=0
cs:000B E81300      call  0021           si 0000   p=1
cs:000E BB0A00      mov   bx,000A        di 0000   a=1
cs:0011 E80D00      call  0021           bp 0000   i=1
cs:0014 BB6400      mov   bx,0064        sp 0014   d=0
cs:0017 E80700
cs:001A BBE803   ds:1004 34 12 D2 04 00 00 00 00 4‡╥♦
cs:001D E80100   ds:100C 00 00 00 00 03 03 00 00     ♥♥
cs:0020▶CC       ds:1014 00 00 B0 48 DF 01 C5 15   ░H▀☺┼§
cs:0021 A10410   ds:101C E8 03 21 00 AD 48 16 32 Φ♥! ¡H▬2

es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺
es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺   ss:0016 0000
es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻      ss:0014▶0000
```

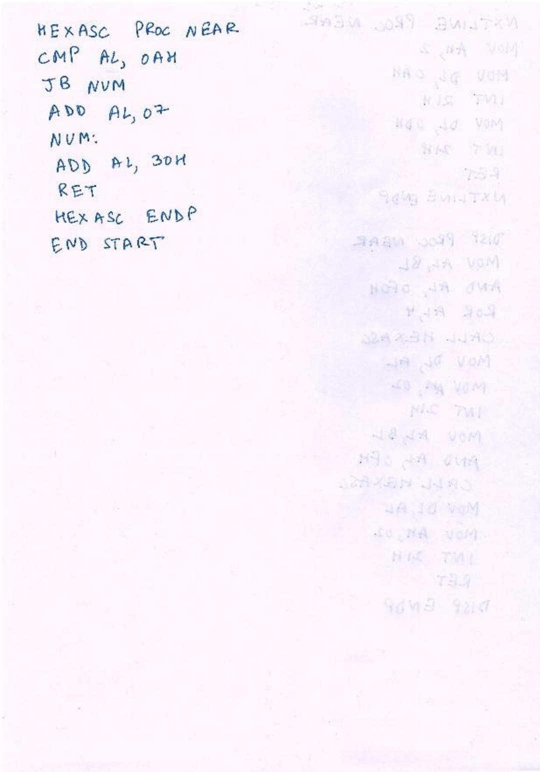
DS: 1004H

Output at 1006H, 1234 in hexadecimal is 4D2

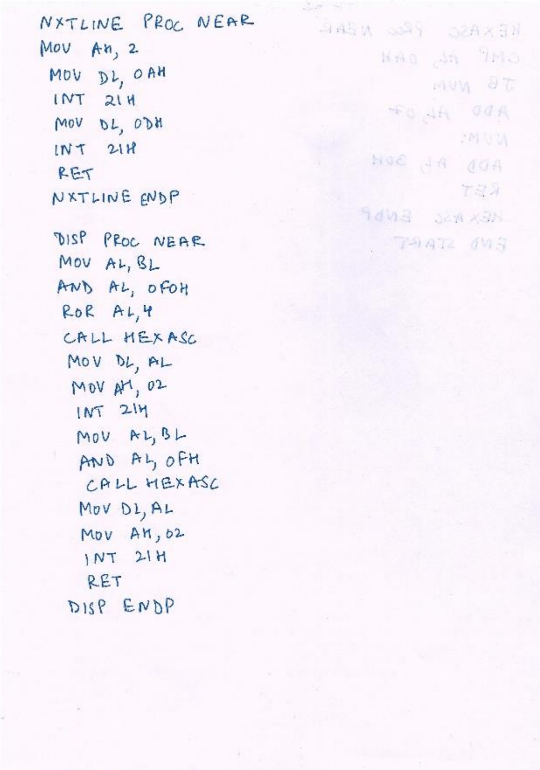
P2: Write a program to input two single-digit hex numbers from keyboard and display their product on the screen.



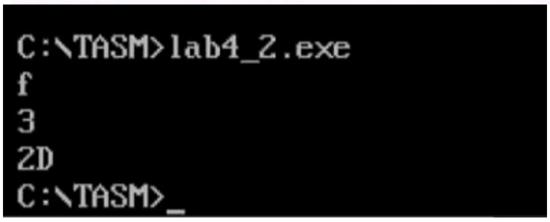
```
P2. .MODEL SMALL
    .STACK 20
    .CODE
    START:
    CALL READKB
    MOV BL, AL
    CALL NXTLINE
    CALL READKB
    MUL BL
    MOV BL, AL
    CALL NXTLINE
    CALL DISP
    MOV AH, 4CH
    INT 21H

    READKB PROC NEAR
    MOV AH, 01
    INT 21H
    CALL ASCHEX
    RET
    READKB ENDP

    ASCHEX PROC NEAR
    CMP AL, 3AH
    JC SUB30
    SUB AL, 07H
    SUB30:
    SUB AL, 30H
    AND AL, 0FH
    RET
    ASCHEX ENDP
```



```
HEXASC PROC NEAR
CMP AL, 0AH
JB NUM
ADD AL, 07
NUM:
ADD AL, 30H
RET
HEXASC ENDP
END START
```



```
NXTLINE PROC NEAR
MOV AH, 2
MOV DL, 0AH
INT 21H
MOV DL, 0DH
INT 21H
RET
NXTLINE ENDP

DISP PROC NEAR
MOV AL, BL
AND AL, 0F0H
ROR AL,4
CALL HEXASC
MOV DL, AL
MOV AH, 02
INT 21H
MOV AL,BL
AND AL, 0FH
CALL HEXASC
MOV DL,AL
MOV AH, 02
INT 21H
RET
DISP ENDP
```

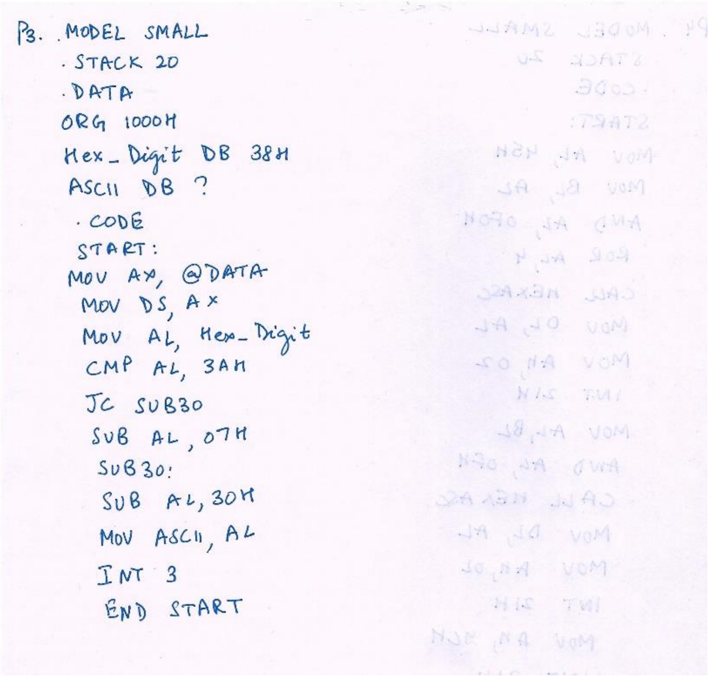


```
C:\TASM>lab4_2.exe
f
3
2D
C:\TASM>_
```

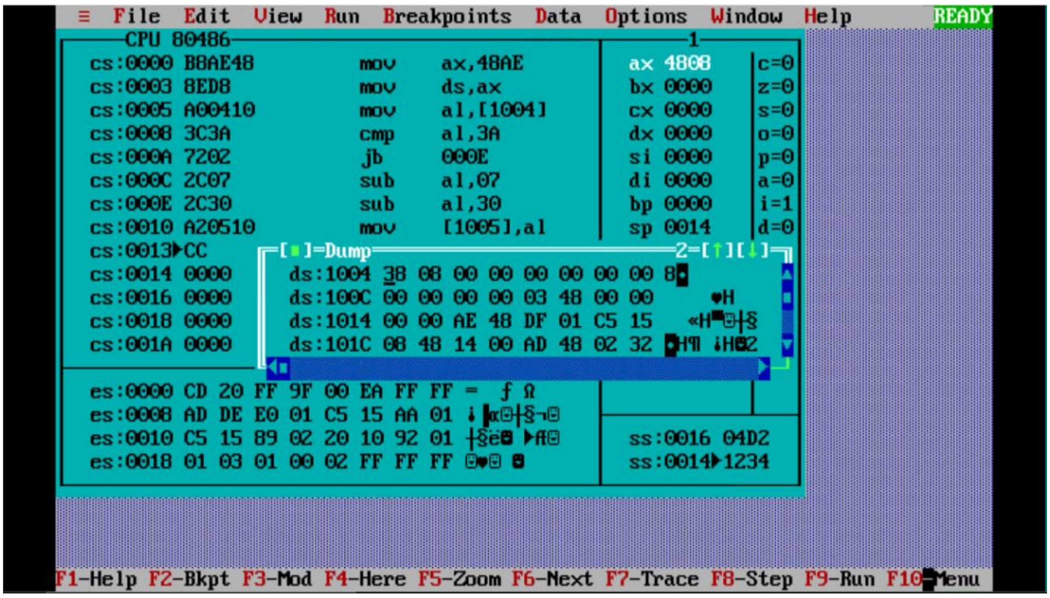
N1 = F ie $(15)_{10}$
N2=3
N1*N2 = 2D ie $(45)_{10}$

Dhruv Maheshwari 2019A7PS0020U

P3: Write a program to convert the given HEXADECIMAL digit to ASCII byte and store the result in memory.



```
P3. .MODEL SMALL
    .STACK 20
    .DATA
    ORG 1000H
    Hex_Digit DB 38H
    ASCII DB ?
     .CODE
     START:
    MOV AX, @DATA
     MOV DS, AX
     MOV AL, Hex_Digit
     CMP AL, 3AH
     JC SUB30
     SUB AL, 07H
      SUB30:
      SUB AL, 30H
      MOV ASCII, AL
      INT 3
       END START
```



≡ File Edit View Run Breakpoints Data Options Window Help READY

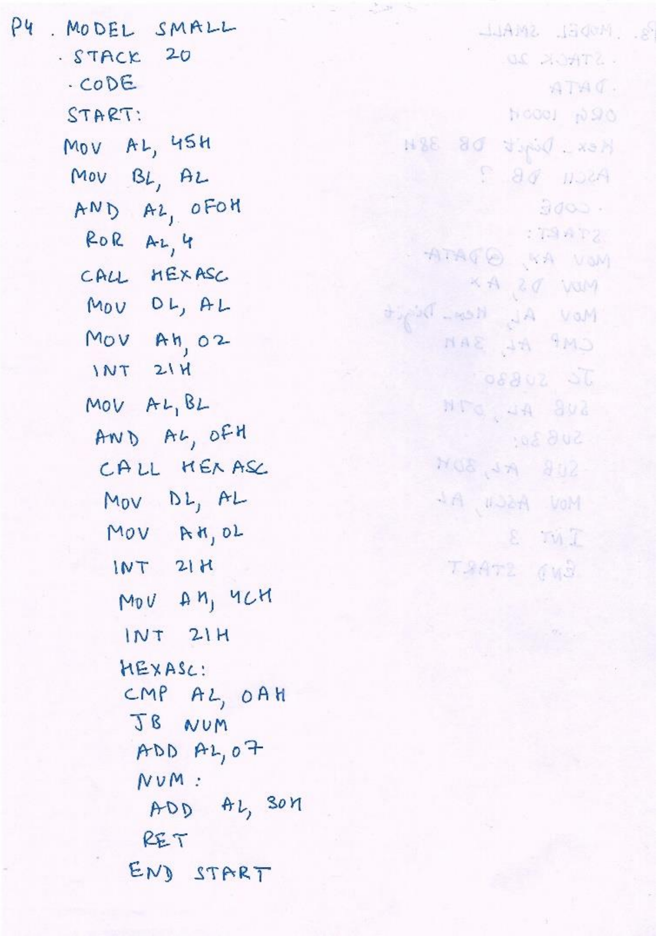
```
CPU 80486                                                   1
cs:0000 B8AE48         mov    ax,48AE         ax 4808  c=0
cs:0003 8ED8           mov    ds,ax           bx 0000  z=0
cs:0005 A00410         mov    al,[1004]       cx 0000  s=0
cs:0008 3C3A           cmp    al,3A           dx 0000  o=0
cs:000A 7202           jb     000E            si 0000  p=0
cs:000C 2C07           sub    al,07           di 0000  a=0
cs:000E 2C30           sub    al,30           bp 0000  i=1
cs:0010 A20510         mov    [1005],al       sp 0014  d=0
cs:0013▶CC
cs:0014 0000     [■]=Dump=====================2=[↑][↓]
cs:0016 0000      ds:1004 38 08 00 00 00 00 00 00 8
cs:0018 0000      ds:100C 00 00 00 00 03 48 00 00   ♥H
cs:001A 0000      ds:1014 00 00 AE 48 DF 01 C5 15   «H▀☺┼§
                  ds:101C 08 48 14 00 AD 48 02 32 ◘H¶ ¡H☻2

es:0000 CD 20 FF 9F 00 EA FF FF  =  ƒ Ω
es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺
es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺      ss:0016 04D2
es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻         ss:0014▶1234
```

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

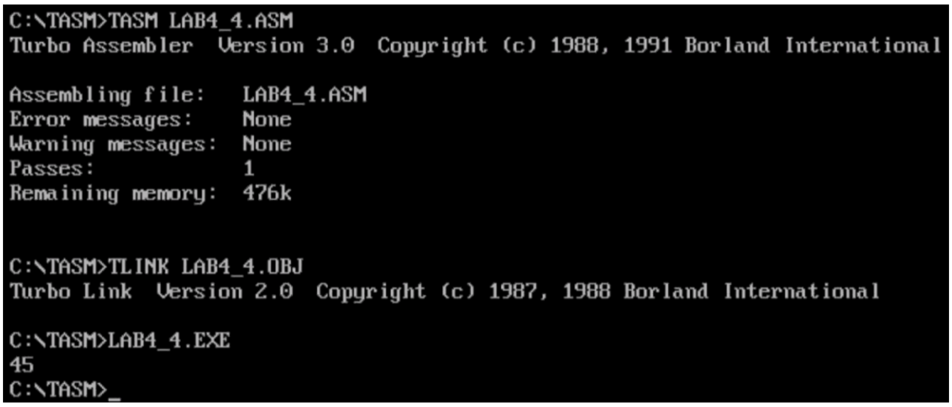
DS:1004H

Output given 08 at 1005

P4: Write a program to display the hexadecimal byte 45H on the screen using DOS interrupts. Previous program is HEXASC (HEX TO ASCII), refer for the HEXASC procedure.



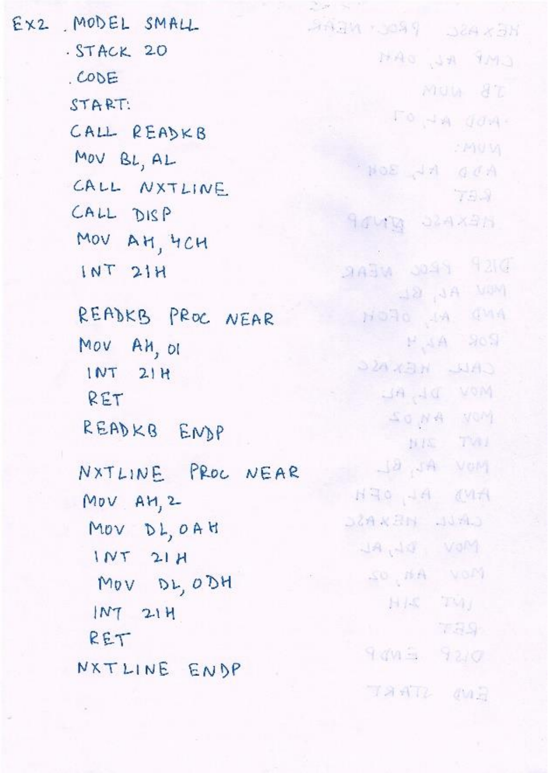
```
P4 . MODEL SMALL
   . STACK 20
   . CODE
   START:
   MOV AL, 45H
   MOV BL, AL
   AND AL, 0F0H
   ROR AL, 4
   CALL HEXASC
   MOV DL, AL
   MOV AH, 02
   INT 21H
   MOV AL, BL
   AND AL, 0FH
   CALL HEXASC
   MOV DL, AL
   MOV AH, 02
   INT 21H
   MOV AH, 4CH
   INT 21H
   HEXASC:
   CMP AL, 0AH
   JB NUM
   ADD AL, 07
   NUM:
    ADD AL, 30H
   RET
   END START
```



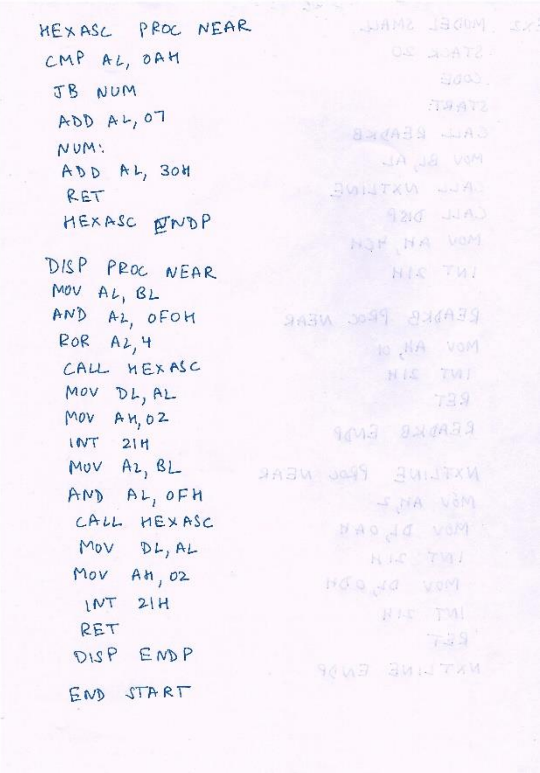
```
C:\TASM>TASM LAB4_4.ASM
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   LAB4_4.ASM
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k


C:\TASM>TLINK LAB4_4.OBJ
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International

C:\TASM>LAB4_4.EXE
45
C:\TASM>_
```

45 displayed on DOS windows

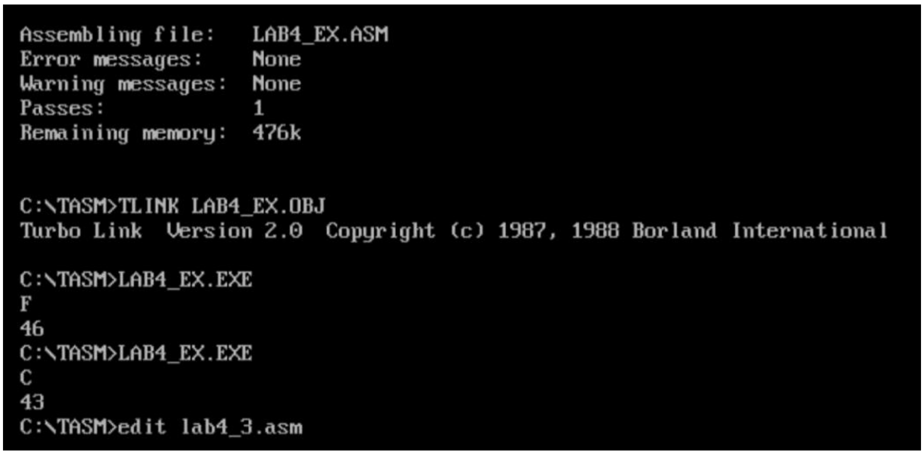
Ex 2: Write a program to accept a character from keyboard and display its ASCII equivalent value on the screen.



```
Ex2 .MODEL SMALL
    .STACK 20
    .CODE
    START:
    CALL READKB
    MOV BL, AL
    CALL NXTLINE
    CALL DISP
    MOV AH, 4CH
    INT 21H

    READKB PROC NEAR
    MOV AH, 01
    INT 21H
    RET
    READKB ENDP

    NXTLINE PROC NEAR
    MOV AH, 2
    MOV DL, 0AH
    INT 21H
    MOV DL, 0DH
    INT 21H
    RET
    NXTLINE ENDP
```



```
HEXASC PROC NEAR
CMP AL, 0AH
JB NUM
ADD AL, 07
NUM:
ADD AL, 30H
RET
HEXASC ENDP

DISP PROC NEAR
MOV AL, BL
AND AL, 0F0H
ROR AL, 4
CALL HEXASC
MOV DL, AL
MOV AH, 02
INT 21H
MOV AL, BL
AND AL, 0FH
CALL HEXASC
MOV DL, AL
MOV AH, 02
INT 21H
RET
DISP ENDP
END START
```



```
Assembling file:   LAB4_EX.ASM
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k


C:\TASM>TLINK LAB4_EX.OBJ
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International

C:\TASM>LAB4_EX.EXE
F
46
C:\TASM>LAB4_EX.EXE
C
43
C:\TASM>edit lab4_3.asm
```

Displays the ASCII value of F ie 46 and C ie 43