Riphah International University

Babar Ali

65731

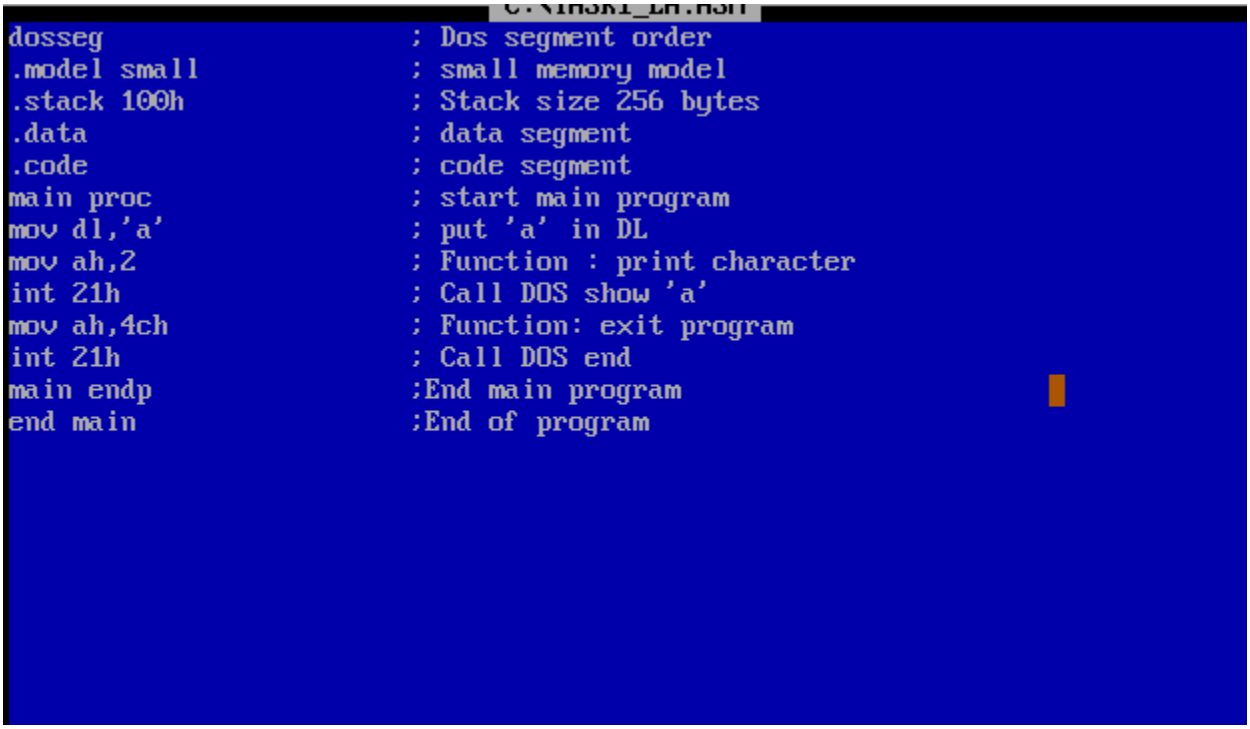
3rd Semester

Computer Organization & AL

Lab 01

Task 1: Print a Single Character Write a MASM program to print a single character on the screen. Example: Print the character 'A'.

The following is the code for the first task.



```
dosseg                  ; Dos segment order
.model small            ; small memory model
.stack 100h             ; Stack size 256 bytes
.data                   ; data segment
.code                   ; code segment
main proc               ; start main program
mov dl,'a'              ; put 'a' in DL
mov ah,2                ; Function : print character
int 21h                 ; Call DOS show 'a'
mov ah,4ch              ; Function: exit program
int 21h                 ; Call DOS end
main endp               ;End main program
end main                ;End of program
```

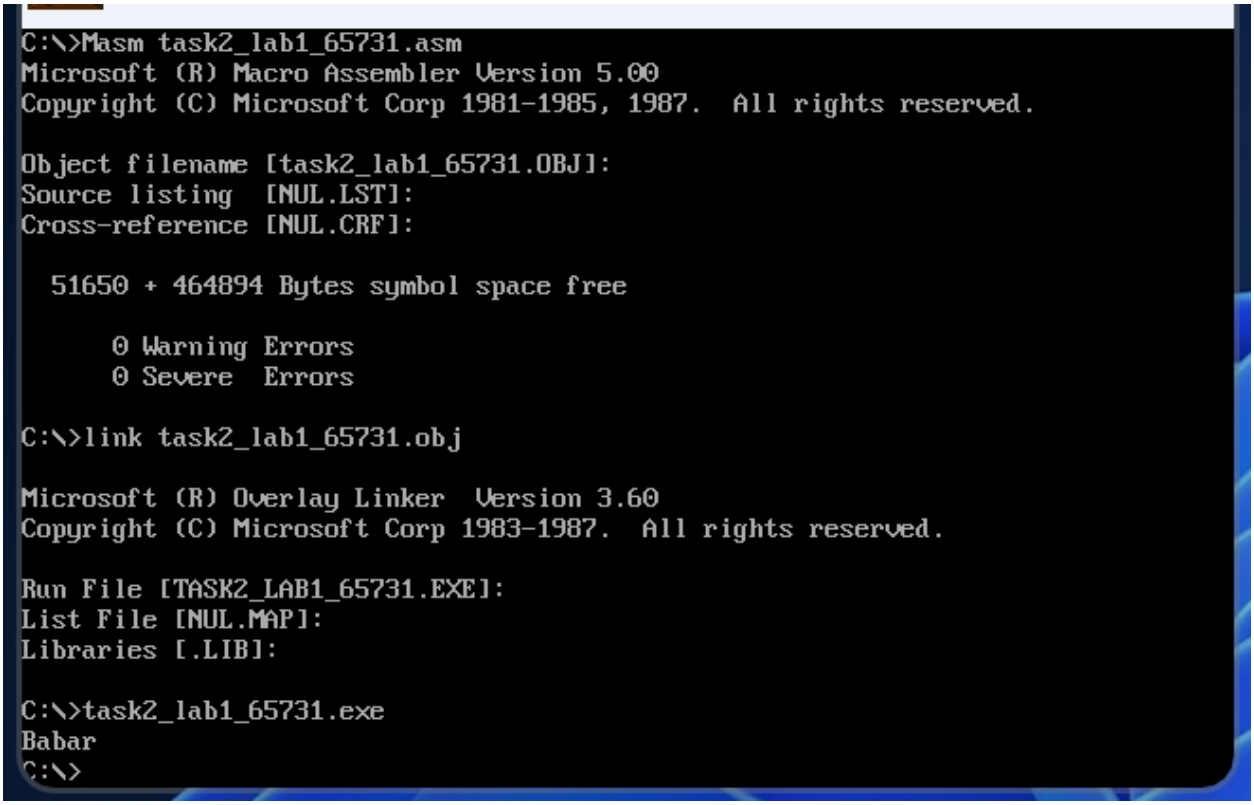
```
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount C C:\8086
Drive C is mounted as local directory C:\8086\

Z:\>C:\

C:\>edit task1_lab1

C:\>edit task1_lab1_65731.asm

C:\>Masm task1_lab1_65731.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [task1_lab1_65731.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51650 + 464894 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>
```

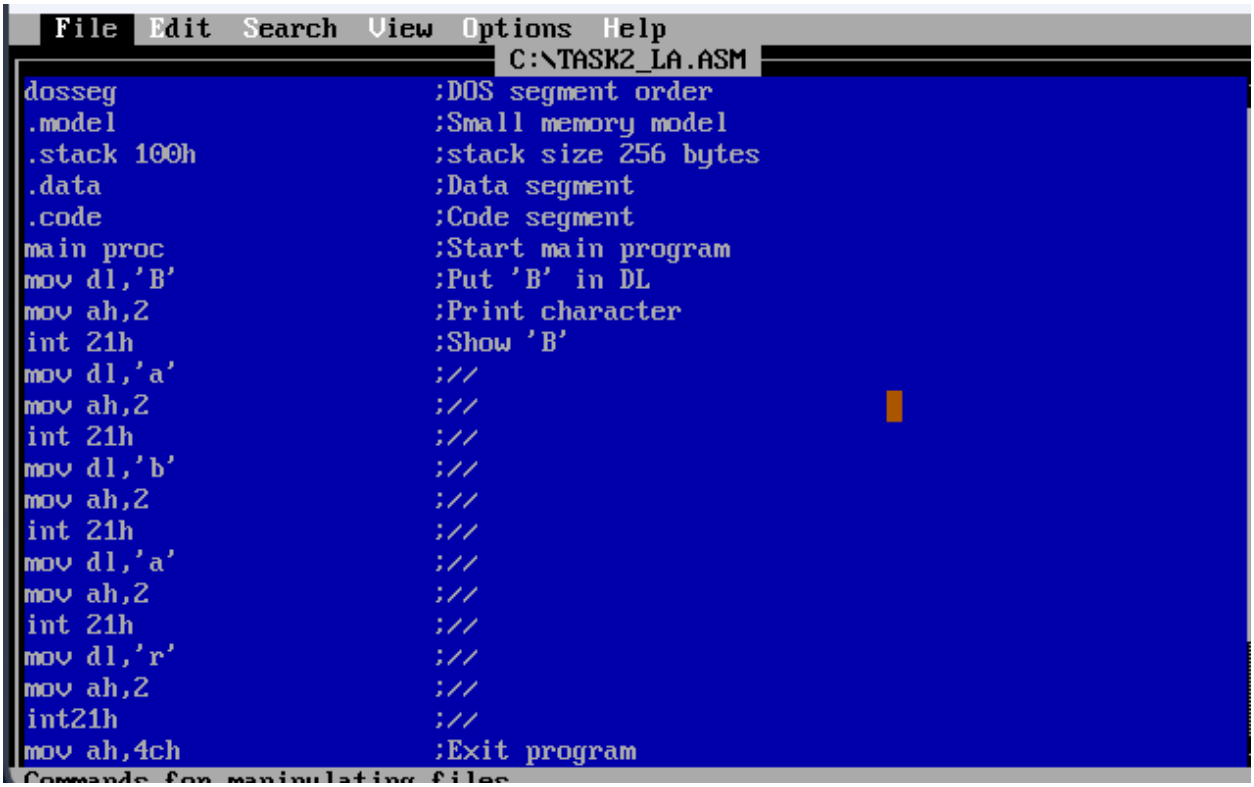
```
C:\>Masm task1_lab1_65731.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [task1_lab1_65731.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51650 + 464894 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link task1_lab1_65731.obj

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

Run File [TASK1_LAB1_65731.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>task1_lab1_65731.exe
a
C:\>
```

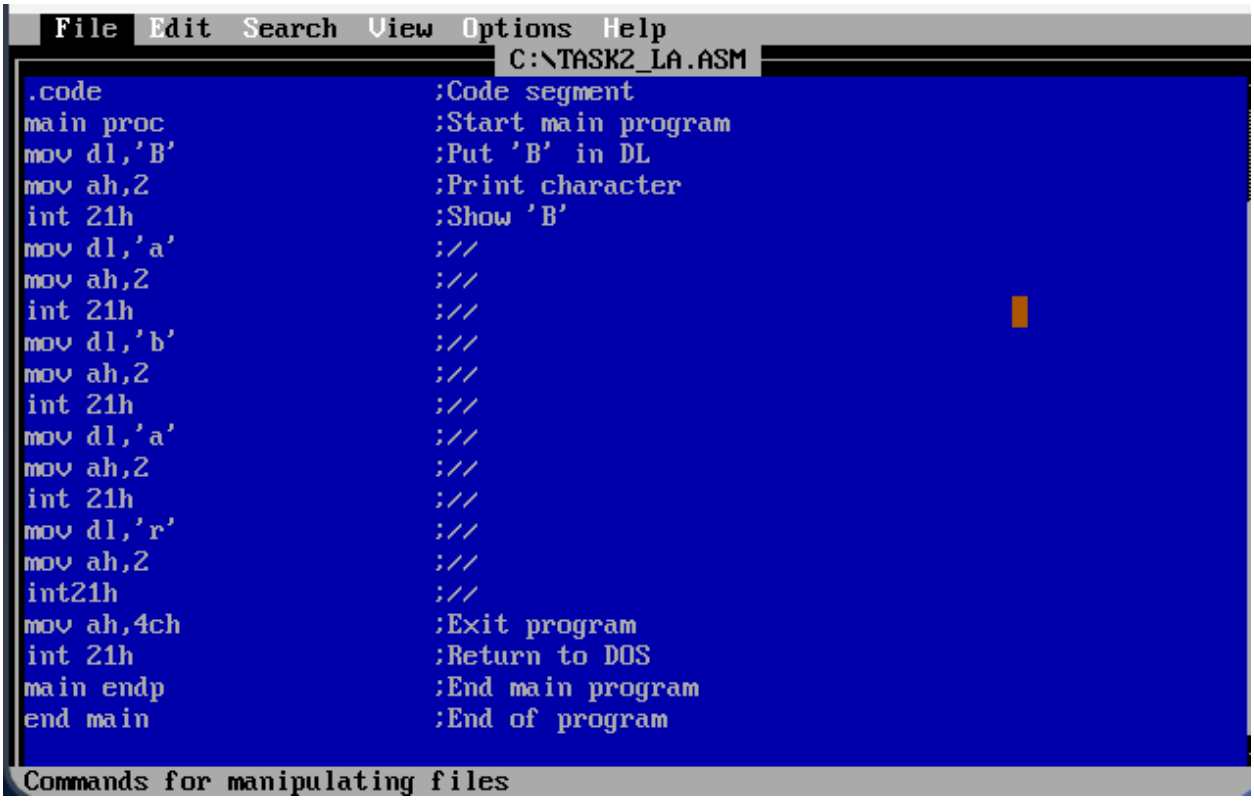
Task 2: Print Your Name Character by Character Write a MASM program that prints your full name by displaying one character at a time.

Example: Print "Mueed" → display 'M', 'u', 'e', 'e', 'd' sequentially.

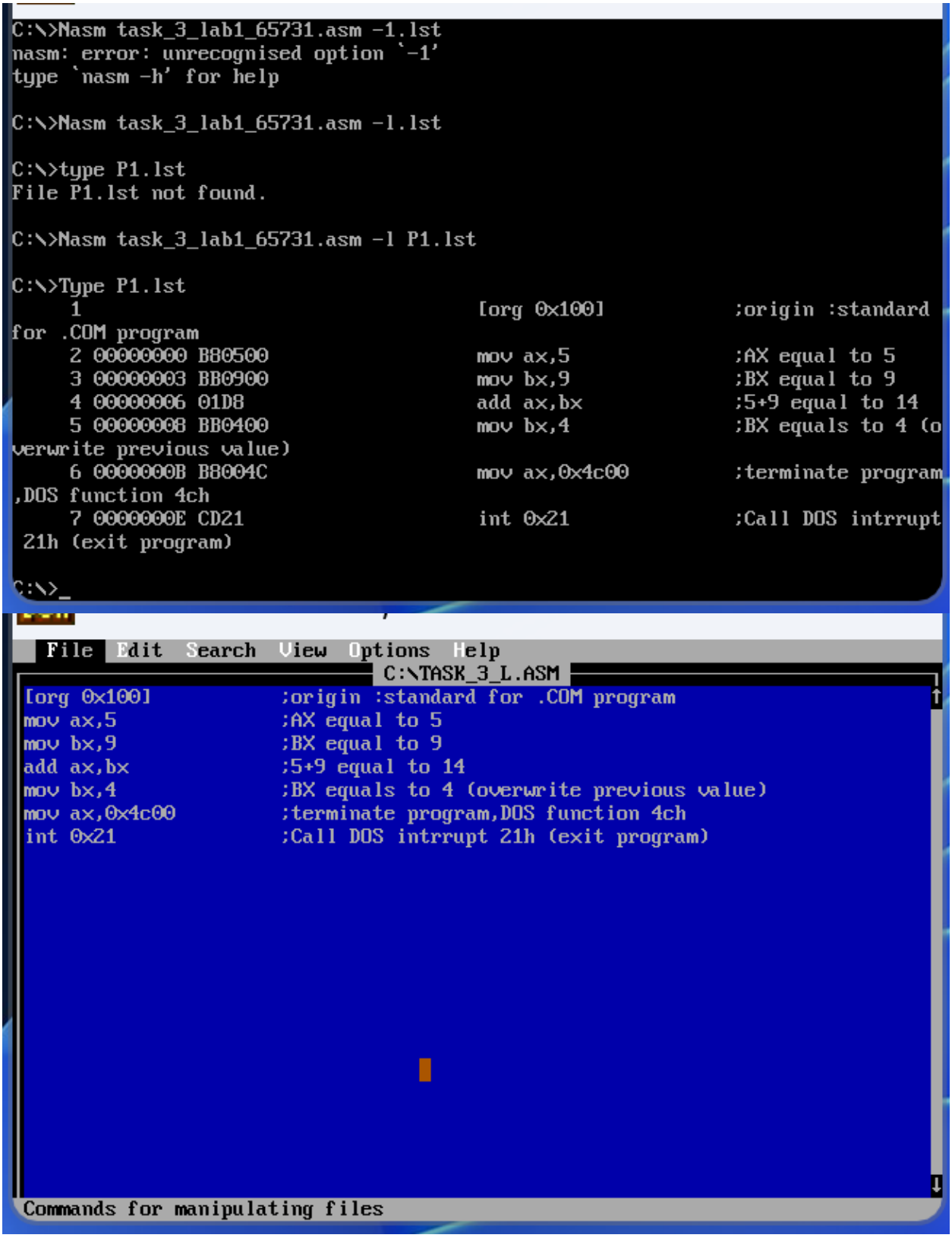
The following is the code for the Second task.




File Edit Search View Options Help
C:\TASK2_LA.ASM
dosseg ;DOS segment order
.model ;Small memory model
.stack 100h ;stack size 256 bytes
.data ;Data segment
.code ;Code segment
main proc ;Start main program
mov dl,'B' ;Put 'B' in DL
mov ah,2 ;Print character
int 21h ;Show 'B'
mov dl,'a' ;//
mov ah,2 ;//
int 21h ;//
mov dl,'b' ;//
mov ah,2 ;//
int 21h ;//
mov dl,'a' ;//
mov ah,2 ;//
int 21h ;//
mov dl,'r' ;//
mov ah,2 ;//
int21h ;//
mov ah,4ch ;Exit program

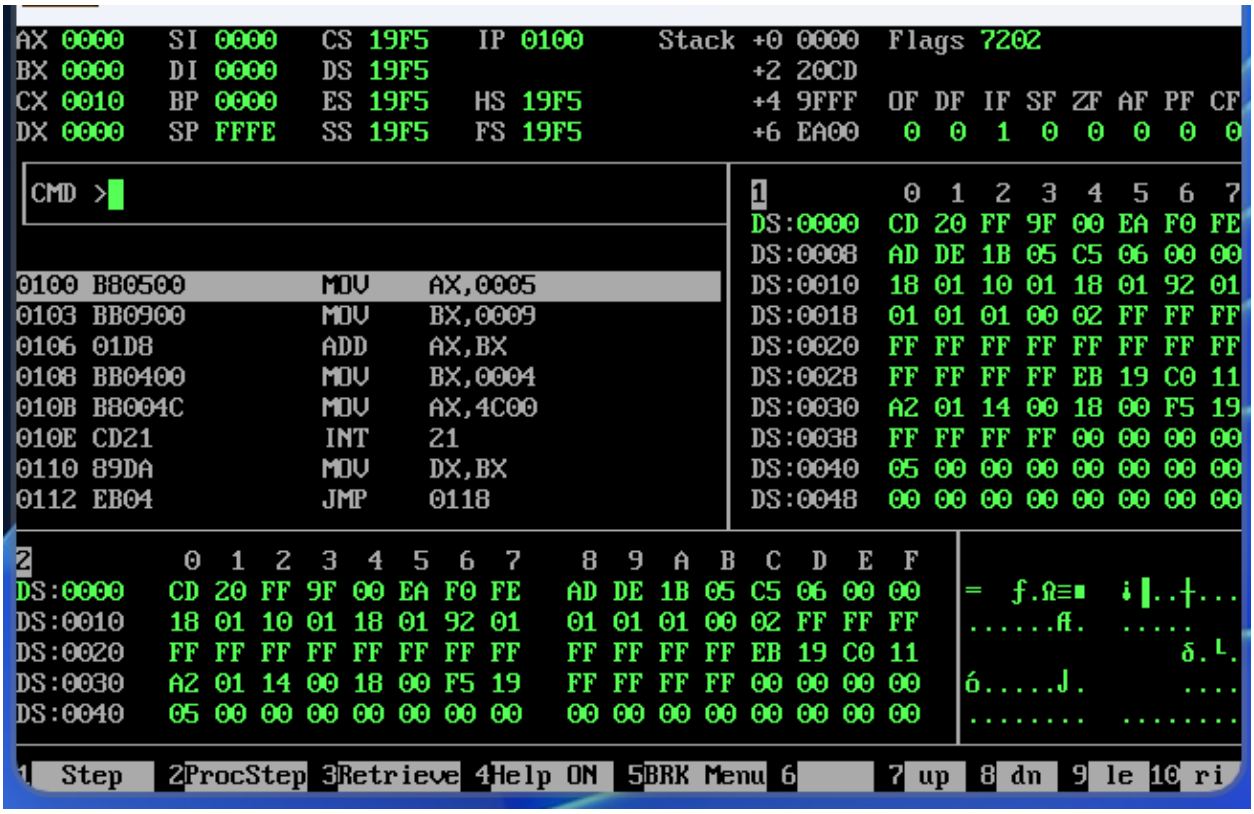

File Edit Search View Options Help
C:\TASK2_LA.ASM
.code ;Code segment
main proc ;Start main program
mov dl,'B' ;Put 'B' in DL
mov ah,2 ;Print character
int 21h ;Show 'B'
mov dl,'a' ;//
mov ah,2 ;//
int 21h ;//
mov dl,'b' ;//
mov ah,2 ;//
int 21h ;//
mov dl,'a' ;//
mov ah,2 ;//
int 21h ;//
mov dl,'r' ;//
mov ah,2 ;//
int21h ;//
mov ah,4ch ;Exit program
int 21h ;Return to DOS
main endp ;End main program
end main ;End of program
Commands for manipulating files

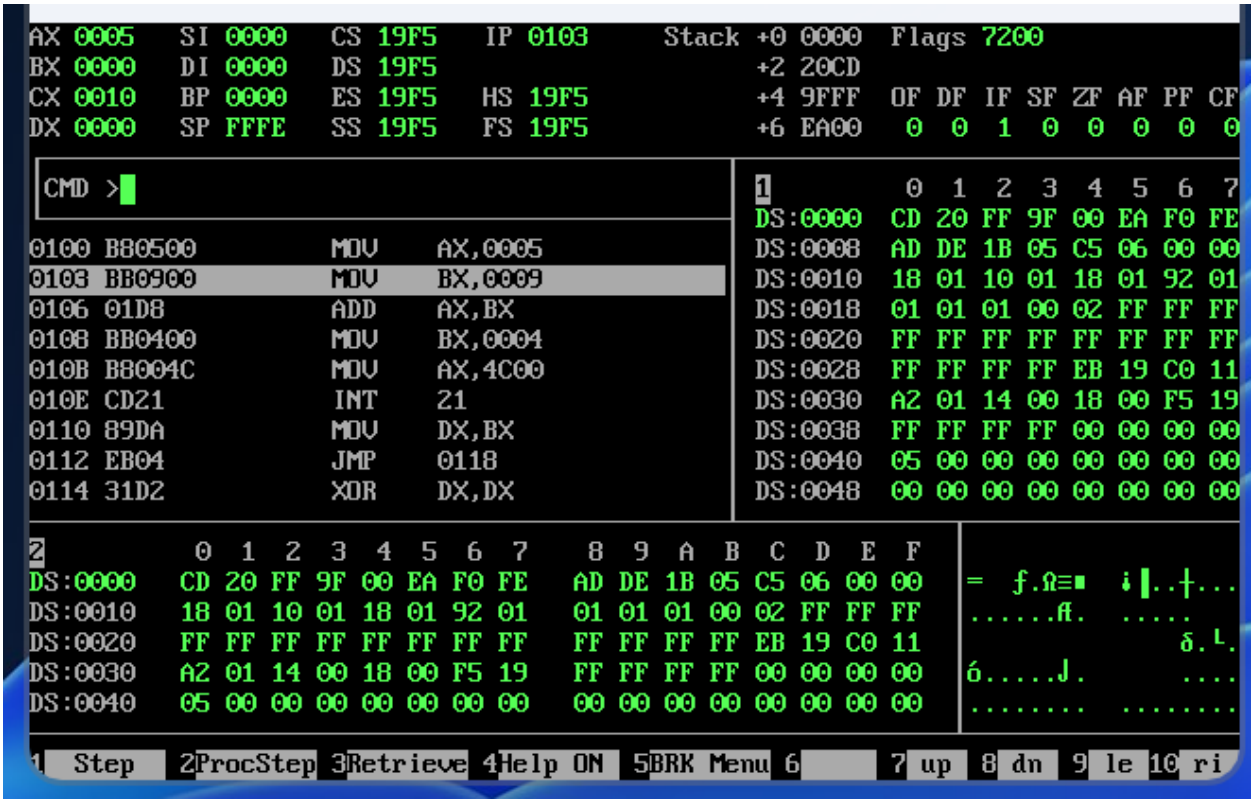
Lab 2: **Task 3:**

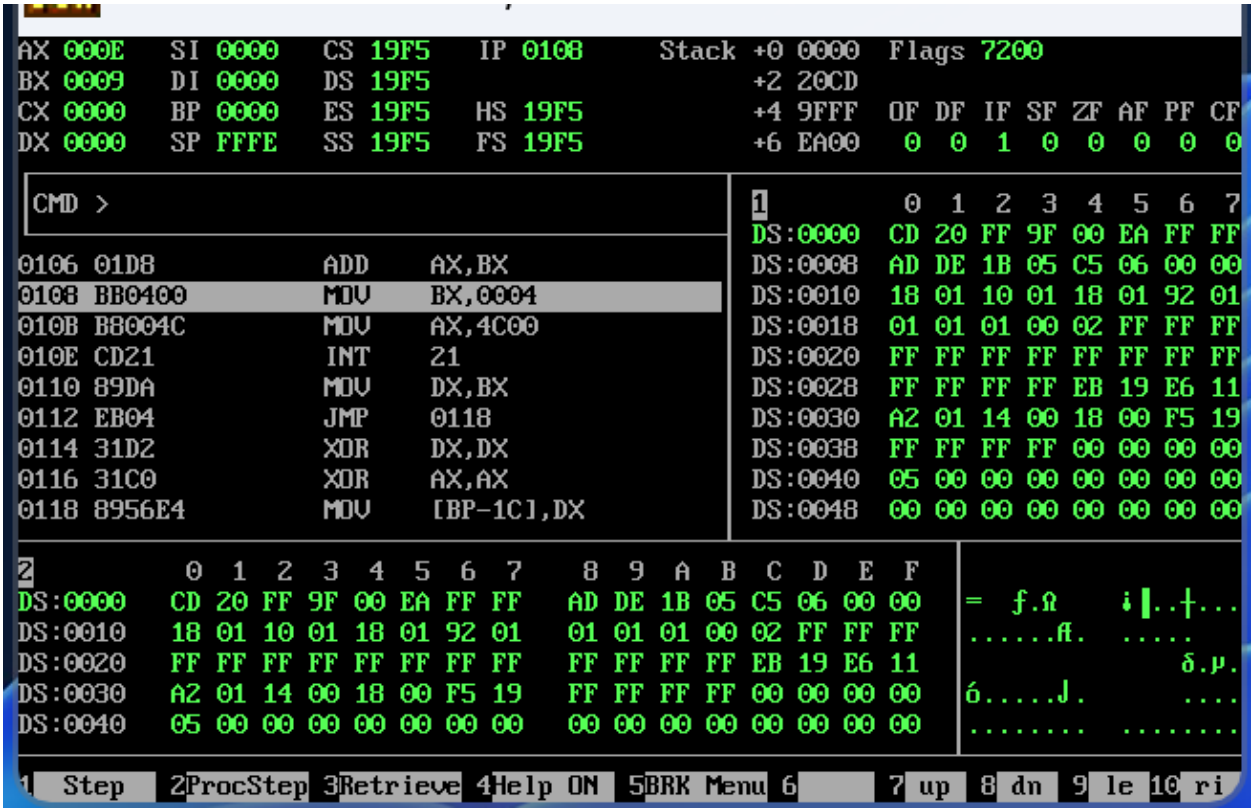


```
C:\>Nasm task_3_lab1_65731.asm -1.lst
nasm: error: unrecognised option `-1'
type `nasm -h' for help

C:\>Nasm task_3_lab1_65731.asm -l.lst

C:\>type P1.lst
File P1.lst not found.

C:\>Nasm task_3_lab1_65731.asm -l P1.lst

C:\>Type P1.lst
     1                                  [org 0x100]             ;origin :standard
for .COM program
     2 00000000 B80500                  mov ax,5                ;AX equal to 5
     3 00000003 BB0900                  mov bx,9                ;BX equal to 9
     4 00000006 01D8                    add ax,bx               ;5+9 equal to 14
     5 00000008 BB0400                  mov bx,4                ;BX equals to 4 (o
verwrite previous value)
     6 0000000B B8004C                  mov ax,0x4c00           ;terminate program
,DOS function 4ch
     7 0000000E CD21                    int 0x21                ;Call DOS intrrupt
 21h (exit program)

C:\>_
```

```
File  Edit  Search  View  Options  Help
                    C:\TASK_3_L.ASM
[org 0x100]             ;origin :standard for .COM program
mov ax,5                ;AX equal to 5
mov bx,9                ;BX equal to 9
add ax,bx               ;5+9 equal to 14
mov bx,4                ;BX equals to 4 (overwrite previous value)
mov ax,0x4c00           ;terminate program,DOS function 4ch
int 0x21                ;Call DOS intrrupt 21h (exit program)

Commands for manipulating files
```


AX 0000 SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0010 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
CMD >
0100 B80500 MOV AX,0005
0103 BB0900 MOV BX,0009
0106 01D8 ADD AX,BX
0108 BB0400 MOV BX,0004
010B B8004C MOV AX,4C00
010E CD21 INT 21
0110 89DA MOV DX,BX
0112 EB04 JMP 0118
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri


AX 0005 SI 0000 CS 19F5 IP 0103 Stack +0 0000 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0010 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
CMD >
0100 B80500 MOV AX,0005
0103 BB0900 MOV BX,0009
0106 01D8 ADD AX,BX
0108 BB0400 MOV BX,0004
010B B8004C MOV AX,4C00
010E CD21 INT 21
0110 89DA MOV DX,BX
0112 EB04 JMP 0118
0114 31D2 XOR DX,DX
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri


AX 000E SI 0000 CS 19F5 IP 0108 Stack +0 0000 Flags 7200
BX 0009 DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
CMD >
0106 01D8 ADD AX,BX
0108 BB0400 MOV BX,0004
010B B8004C MOV AX,4C00
010E CD21 INT 21
0110 89DA MOV DX,BX
0112 EB04 JMP 0118
0114 31D2 XOR DX,DX
0116 31C0 XOR AX,AX
0118 8956E4 MOV [BP-1C],DX
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA FF FF
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri