

COAL (Computer Organization & Assembly Language)

Lab Task-1

**Submitted by: Ahmed Sameer Hashemy
Sap ID: 66026**

**Submitted to: Mr. Mueed Ahmed
BSAI 3-1 Batch Adviser**

Part A – MASM Task

Tools: DOSBox, MASM, NASM

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

Solution:

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program... File Edit Search View Options Help C:\ASH_L1T1.ASM

```
dosseg          ;Organize segments in standard DOS order
.model small    ;Memory model
.stack 100h     ;Allocate 256 bytes (100h) of stack space
.data           ;Needed for structure
.code           ;start of code
main proc       ;Start the main procedure
  mov dl,'A'    ;Load ASCII value of 'A'
  mov ah,2       ;set ah equal to 2
  int 21h        ;interrupt 21h (prints "A")
  mov ah,4ch     ;Dos function to terminate program
  int 21h        ;exit to DOS
main endp       ;End of procedure
end main        ;Tell assembler where program execution starts
```

F1=Help | Line:15 Col:73

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program... C:\>edit ASH_L1T2.asm
C:\>edit ASH_L1T1.asm 1st
C:\>edit ASH_L1T2.asm 2nd
C:\>masm ASH_L1T2.asm; Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51690 + 464854 Bytes symbol space free
0 Warning Errors
0 Severe Errors
C:\>link ASH_L1T2.obj; 3rd
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

C:\> ASH_L1T2.exe 4th
SAMEER
C:\>S_

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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 1, Program... File Edit Search View Options Help C:\ASH_L1T2.ASM

```
dosseg
.model small
.stack 100h
.data
.code
main proc
mov dl,'S'
mov ah,2
int 21h
mov dl,'A'
mov ah,2
int 21h
mov dl,'M'
mov ah,2
int 21h
mov dl,'E'
mov ah,2
int 21h
mov dl,'E'
mov ah,2
int 21h
mov dl,'R'
mov ah,2
int 21h
F1=Help
```

Line:20 Col:76

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 1, Program... File Edit Search View Options Help C:\ASH_L1T2.ASM

```
mov ah,2
int 21h
mov dl,'A'
mov ah,2
int 21h
mov dl,'M'
mov ah,2
int 21h
mov dl,'E'
mov ah,2
int 21h
mov dl,'E'
mov ah,2
int 21h
mov dl,'R'
mov ah,2
int 21h
mov ah,4ch
int 21h
main endp
end main
```

F1=Help

Line:27 Col:76

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

C:\>edit ASH_L1T2.asm
C:\>edit ASH_L1T1.asm
C:\>edit ASH_L1T2.asm 1st
C:\>masm ASH_L1T2.asm; 2nd
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51690 + 464854 Bytes symbol space free

0 Warning Errors
0 Severe Errors
C:\>link ASH_L1T2.obj; 3rd
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

C:\> ASH_L1T2.exe 4th
SAMEER
C:\>S_

Part B – NASM Task

Task 3: Display Register Values Using AFD

- Write a NASM program to display the contents of CPU registers (e.g., AX, BX, and CX) using DOS interrupt INT 21h the AFD (ASCII Formatted Display) technique.

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

File Edit Search View Options Help C:\ASH_L2T1.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 equals to 14
mov bx,4          ;BX equals to 4 (overwrites previous value)
mov ax,0x4c00     ;Terminate program, DOS function 4ch
int 0x21          ;Call DOS interrupt 21h (exit program)
```

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

```

Z:\>c:
C:\> edit ASH_L2T1.asm
C:\>Nasm ASH_L2T1.asm -o p1.com
C:\>p1.com
C:\>Nasm ASH_L2T1.asm -l.1st
nasm: error: unrecognised option '-l'
type 'nasm -h' for help
C:\>Nasm ASH_L2T1.asm -l.1st
C:\>type p1.1st
1 [org 0x100]
2 00000000 B80500
3 00000003 BB0100
4 00000006 01D8
5 00000008 BB0400
6 0000000B B8004C
7 0000000E CD21
        mov ax,8
        mov bx,1
        add ax,bx
        mov bx,4
        mov ax,0x4c00
        int 0x21

C:\>Afd p1.com

```

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

	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 0

CMD >	1	0 1 2 3 4 5 6 7
0100 B80500	MOV AX,0005	DS:0000 CD 20 FF 9F 00 EA F0 FE
0103 BB0900	MOV BX,0009	DS:0008 AD DE 1B 05 C5 06 00 00
0106 01D8	ADD AX,BX	DS:0010 18 01 10 01 18 01 92 01
0108 BB0400	MOV BX,0004	DS:0018 01 01 01 00 02 FF FF FF
010B B8004C	MOV AX,4C00	DS:0020 FF FF FF FF FF FF FF FF
010E CD21	INT 21	DS:0028 FF FF FF EB 19 C0 11
0110 89DA	MOV DX,BX	DS:0030 A2 01 14 00 18 00 F5 19
0112 EB04	JMP 0118	DS:0038 FF FF FF 00 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	= f.Ω≡■ i .+...
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 FFf.
DS:0020	FF FF FF FF FF FF FF	FF FF FF EB 19 C0 11	δ. L.
DS:0030	A2 01 14 00 18 00 F5 19	FF FF FF FF 00 00 00 00	δ.....J.
DS:0040	05 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00

1 Step | 2 ProcStep | 3 Retrieve | 4 Help ON | 5 SBRK Menu | 6 | 7 up | 8 dn | 9 le | 10 ri

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 1, Program... — X

AX 0005	SI 0000	CS 19F5	IP 0106	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 0

CMD >	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
0103 BB0900	MOV BX,0009										
0106 01D8	ADD AX,BX										
0108 BB0400	MOV BX,0004										
010B BB004C	MOV AX,4000										
010E CD21	INT 21										
0110 89DA	MOV DX,BX										
0112 EB04	JMP 0118										
0114 31D2	XOR DX,DX										
0116 31C0	XOR AX,AX										

2 0 1 2 3 4 5 6 7 8 9 A B C D E F	= f.Ω i .+...A. δ. p. 6.....J.										
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 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 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 ? up 8 dn 9 le 10 ri

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program... — X

AX 000E	SI 0000	CS 19F5	IP 0108	Stack +0 0000	Flags 7200
BX 0009	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 0

CMD >	1 0 1 2 3 4 5 6 7	DS:0000 CD 20 FF 9F 00 EA FO 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
0106 01D8	ADD AX,BX										
0108 BB0400	MOV BX,0004										
010B BB004C	MOV AX,4000										
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										

2 0 1 2 3 4 5 6 7 8 9 A B C D E F	= f.Ω≡■ i .+...A. δ. L. 6.....J.										
DS:0000 CD 20 FF 9F 00 EA FO 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 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 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 ? up 8 dn 9 le 10 ri

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program... — X

AX 4C00	SI 0000	CS F000	IP 14A0	Stack +0 42BD	Flags 7000
BX 0004	DI 0000	DS 19F5		+2 06C5	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 7000	OF DF IF SF ZF AF PF CF
DX 0000	SP FFF2	SS 19F5	FS 19F5	+6 0110	0 0 0 0 0 0 0 0 0

```

CMD >[red arrow pointing here]
010E CD21      INT    21
14A0 FB        STI
14A1 FE        DB     FE
14A2 3825      CMP    [DI],AH
14A4 00CF      ADD    BH,CL
14A6 CB        RET    Far
14A7 51        PUSH   CX
14A8 B94001    MOV    CX,0140
14AB E2FE      LOOP   14AB

```

1	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	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 EB 19 E6 11	DS:0030 A2 01 14 00 18 00 F5 19	DS:0038 FF FF FF 00 00 00 00 00	DS:0040 05 00 00 00 00 00 00 00	DS:0048 00 00 00 00 00 00 00 00
Z	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	01 01 01 00 02 FF FF FF	FF FF FF FF EB 19 E6 11	FF FF FF FF 00 00 00 00	FF FF FF FF 00 00 00 00	A2 01 14 00 18 00 F5 19	05 00 00 00 00 00 00 00	= f.Ω i .+...fl.	δ.μ. 6.....J.
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	DS:0030 A2 01 14 00 18 00 F5 19	DS:0040 05 00 00 00 00 00 00 00	DS:0048 00 00 00 00 00 00 00 00	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	
DS:0020	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	DS:0040 05 00 00 00 00 00 00 00	DS:0048 00 00 00 00 00 00 00 00	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	DS:0028 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	DS:0048 00 00 00 00 00 00 00 00	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	DS:0028 FF FF FF EB 19 E6 11	DS:0030 A2 01 14 00 18 00 F5 19	
DS:0040	05 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	DS:0048 00 00 00 00 00 00 00 00	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	DS:0028 FF FF FF EB 19 E6 11	DS:0030 A2 01 14 00 18 00 F5 19	DS:0038 FF FF FF 00 00 00 00 00	

1 Step 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 7 up 8 dn 9 le 10 ri

- **CPU Registers** (AX, BX, etc.) and their current values.
- **Disassembled Code** your program's instructions like MOV, ADD, and INT 21 h.
- **Memory Dump** raw bytes in memory (both hex + ASCII).
- **Execution Control** you can step through code, see how each instruction changes registers.