

**Name:** Muhammad Hamza

**Class:** BS (AI)

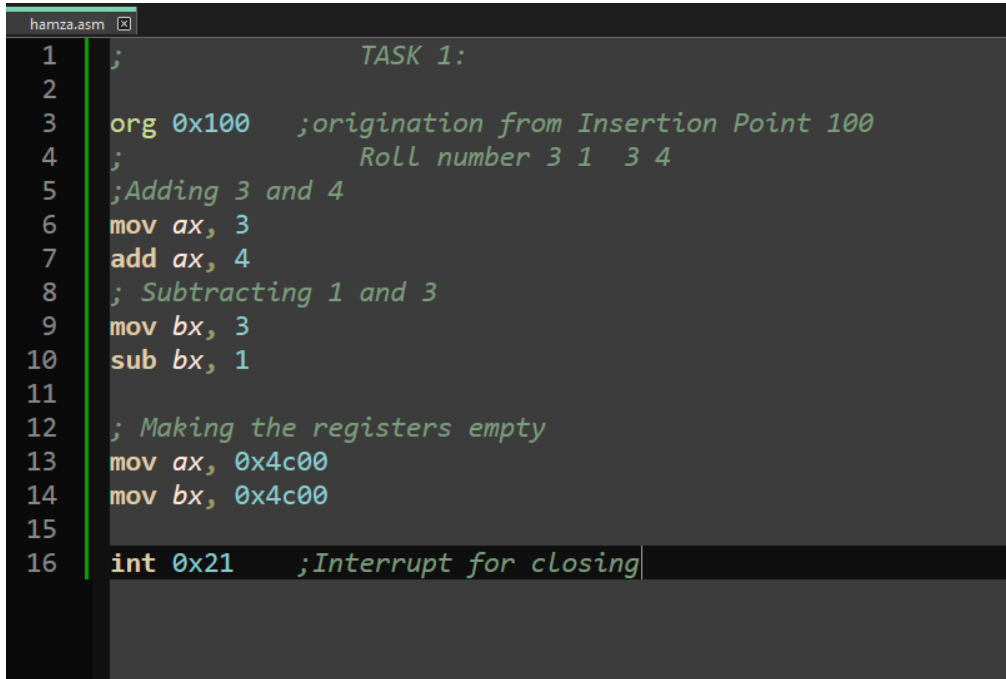
**Section:** 4A

**Roll number:** 22F-3134

### **COAL LAB 2:**

**Task:** Addition of first and last number and subtraction of middle two numbers in roll number '3134'.

Code file:

A screenshot of a text editor window titled 'hamza.asm'. The editor contains assembly code for a task. The code is as follows:

```
1 ; TASK 1:
2
3 org 0x100 ;origination from Insertion Point 100
4 ; Roll number 3 1 3 4
5 ;Adding 3 and 4
6 mov ax, 3
7 add ax, 4
8 ; Subtracting 1 and 3
9 mov bx, 3
10 sub bx, 1
11
12 ; Making the registers empty
13 mov ax, 0x4c00
14 mov bx, 0x4c00
15
16 int 0x21 ;Interrupt for closing
```

### **Steps:**

- Starting from 'org' (origination) from IP (Insertion Pointer) 100.
- Initializing AX and BX registers with 'mov' statement.
- Addition of 3 and 4 in 'AX' register, and subtraction of 3 and 1 in 'BX' register.
- Then initializing the registers 0x4c00, to terminate the program.

- Then finally terminating the program using interrupt 0x21.

## DOS Box:

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com

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

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

Z:\>X:

X:\>nasm hamza.asm -o hamza.com

X:\>afd hamza.com

AFD-Pro is done

X:\>nasm hamza.asm -o hamza.com

X:\>afd hamza.com

AFD-Pro is done

X:\>_
```

## Steps:

- Mounting X variable in Local Disk C, NASM folder.
- In the variable 'X', using 'nasm' command with file name, using "-o" for cycles, core of our computer system.
- After that using '*afd debugger*' with our file name (hamza.com).

## Registers in the start:



Adding first and last numbers of roll number '3134', the value  $3 + 4 = 7$  will be stored in register 'AX'.

Subtracting 2<sup>nd</sup> and 3<sup>rd</sup> number of roll number '3134', the value  $3 - 1 = 2$  will be stored in 'BX'.



Operations performed on register 'AX' and 'BX':

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0007 SI 0000 CS 19F5 IP 010D Stack +0 0000 Flags 7200
BX 0002 DI 0000 DS 19F5 +2 20CD
CX 0015 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 >

0109 81EB0100 SUB BX,0001
010D B8004C MOV AX,4C00
0110 BB004C MOV BX,4C00
0113 CD21 INT 21
0115 D2 DB D2
0116 31C0 XOR AX,AX
0118 8956E4 MOV [BP-1C],DX
011B 8946E6 MOV [BP-1A],AX
011E C746F60000 MOV [BP-0A],0000

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 DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.n= i | .t...
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....f. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 .....L.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 6.....J. ....
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
```

Terminating program:

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

AX 0000	SI 0000	CS 19F5	IP 0100	Stack +0 0000	Flags 7202
BX 0000	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 >

Program terminated OK

0100 B80300	MOV	AX,0003
0103 050400	ADD	AX,0004
0106 BB0300	MOV	BX,0003
0109 81EB0100	SUB	BX,0001
010D B8004C	MOV	AX,4C00
0110 BB004C	MOV	BX,4C00
0113 CD21	INT	21
0115 D2	DB	D2

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	E4	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	= f.Ω ÷ ..†...
DS:0010	18	01	10	01	18	01	92	01	01	01	01	00	02	FF	FF	FF	.....ff. ....
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	E4	11	δ.Σ.
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00	6.....J. ....
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

Making a '.lst' file:

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com

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

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

Z:\>C:

C:\>nasm plus.asm -o abc.com

C:\>nasm plus.asm -l pluslist.lst

C:\>type pluslist.lst
1                               [org 0x100]
2 00000000 B80300               mov ax,3
3 00000003 B80200               mov bx,2
4 00000006 01DB                 add ax,bx
5 00000008 B8004C               mov ax,0x4c00
6 0000000B CD21                 int 21h

C:\>_
```

- Addresses are incrementing by 3 and 2.
- Origination with 00000000
- Initializing AX and BX with 3 and 2 respectively.
- Then at 00000008, for printing out.
- And then CD21 for termination of program.