



Computer Organization & Assembly Language (0+1)

Assignment

Fall 2021

Students Name: AHMED ALI

ANSARI

Students ID: 1402-2020

Batch: 2020B

ASSIGNMENT

Important Notes: ZERO Tolerance for plagiarism and cheating if found Assignment will be marked as “ZERO”

CLO-2: Evaluate different technique to gain Assembly language core knowledge.	(C3, PO-3))
--	--------------------

Submission Guidelines:**Marks: 10**

- Hard copy + Soft copy should be submitted + asm. File should also be submitted to google classroom
- Assignment should be in Pdf file.

Your answer must follow the below given specifications.

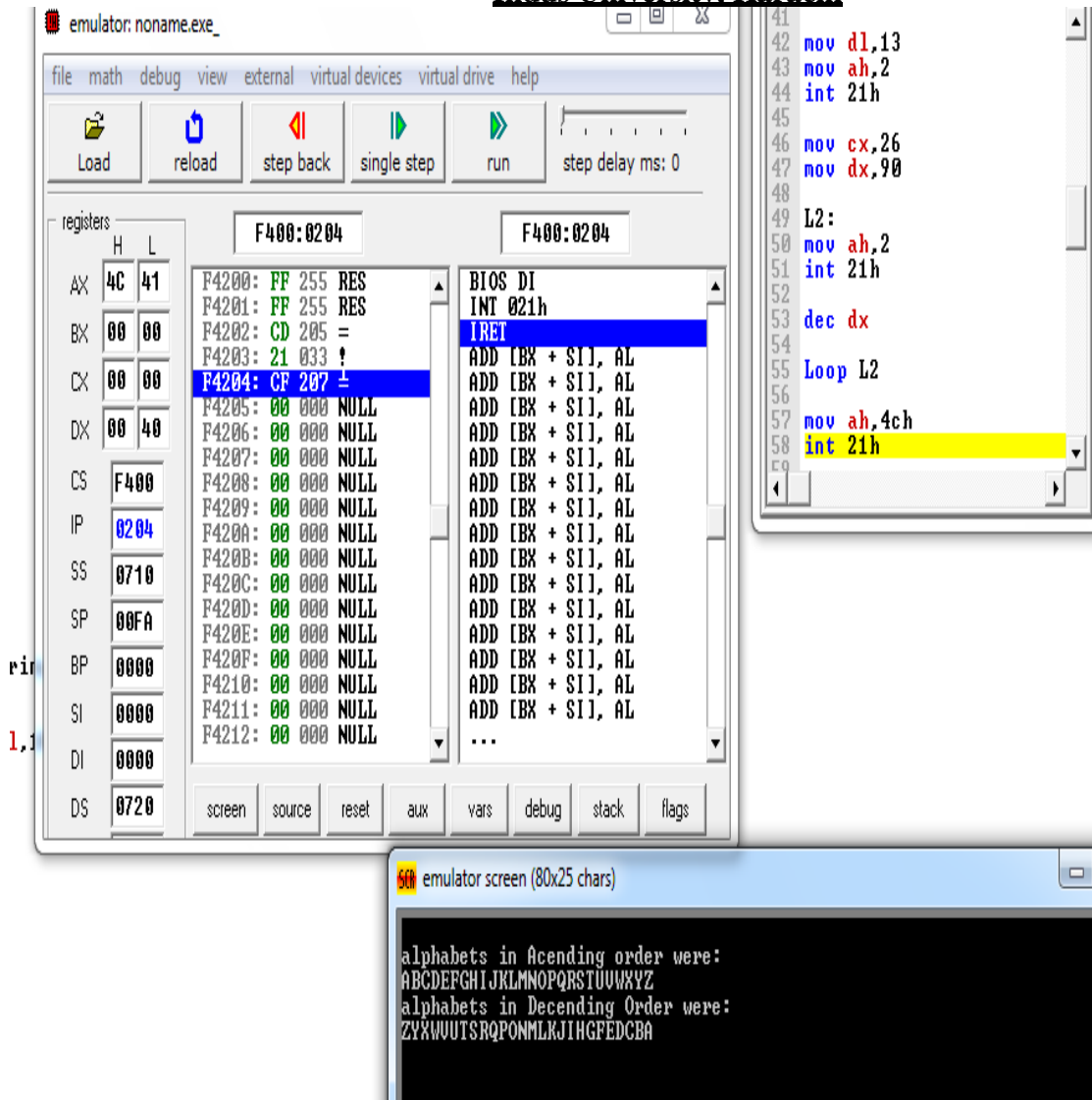
- Font style: “Times New Roman”
- Font color: “Black”
- Font size: “12”
- **Bold** for heading only.

Q1: Construct the program that increment and decrement the values store in 8–bit register by using LEA function also implement carriage return and offset message in your program.

```

01 .model small
02 .stack 100h
03 .data
04
05 string1 db 0AH,0DH, 'alphabets in Ascending order were : $'
06 string2 db 0AH,0DH, 'alphabets in Decending Order were :$'
07 .code
08 main proc
09     mov ax,@data
10     mov ds,ax
11     mov ah,9
12     lea dx,string1
13     int 21h
14
15     mov dl,10
16     mov ah,2
17     int 21h
18
19     mov dl,13
20     mov ah,2
21     int 21h
22
23     mov cx,26
24     mov dx,65
25
26 L1:
27     mov ah,2
28     int 21h
29
30     inc dx
31
32 Loop L1
33     mov ax,@data
34     mov ds,ax
35         mov dx,offset string2
36         mov ah,09h
37         int 21h
38             mov dl,10
39     mov ah,2
40     int 21h
41
42     mov dl,13
43     mov ah,2
44     int 21h
45
46     mov cx,26
47     mov dx,90
48
49 L2:
50     mov ah,2
51     int 21h
52
53     dec dx
54
55 Loop L2
56
57     mov ah,4ch
58     int 21h
59
60     main endp
61 end main

```



Q2: Construct a program that utilizes a loop to compute the first seven values of the Fibonacci number series 1,1,2,3,5,8,13, where $F_n = F_{n-1} + F_{n-2}$ is the rule. The byte storage array named Fibonacci saves the remaining five items in the Fibonacci sequence in the memory exactly the same array.

```
.model small
.org 100h
.data
ANS DW?
string1 db 0AH,0DH, 'fibonacci series : $'

.code
main proc |
    mov ax, @data
    mov ds, ax
    mov dl, 1
    mov bl, 0
    mov cl, 8

    SUM :
    Add bl, dl
    mov al, bl
    mov bl, dl
    mov dl, al
    LOOP SUM

    sub dl, 1
    mov ANS, dx

    mov ah, 4ch
    int 21h

main endp
end main
```

Q3: Write a program which ask user to enter a digit from (0 to 9 only) and then print the table of that number.

The program look like this

Please enter a digit = 3

Table of 3 will be print

```

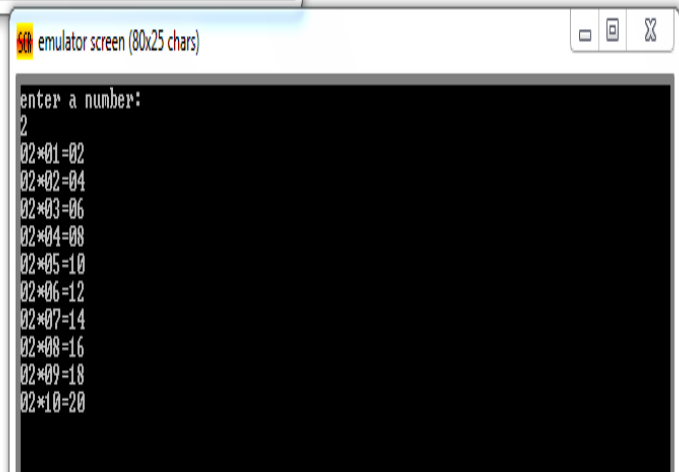
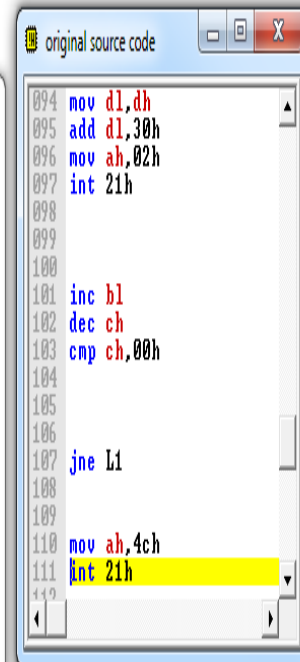
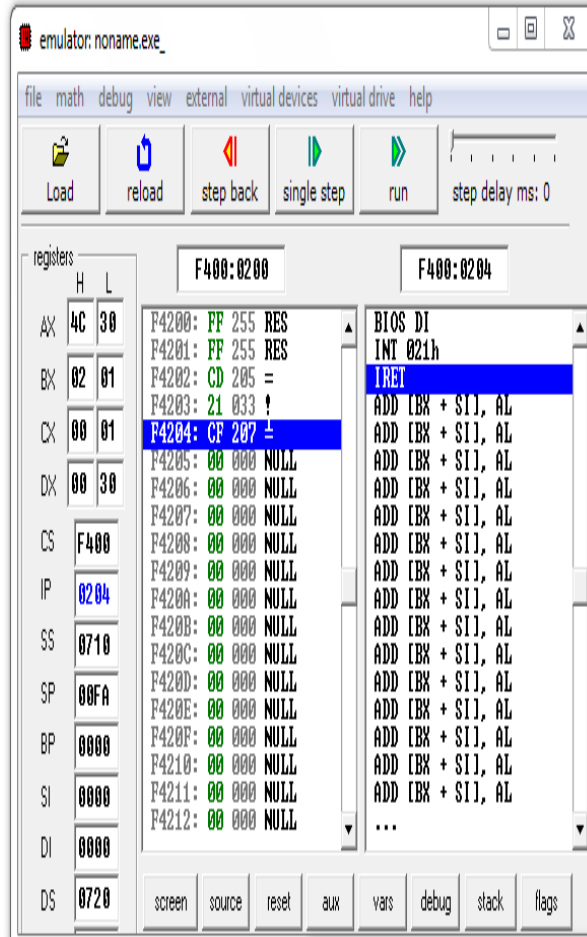
0001 .model small
0002 .stack 100h
0003 .data
0004 A DB 'enter a number : $'
0005 .code
0006 main proc
0007
0008     mov ax,@data
0009     mov ds,ax
0010     lea dx,A
0011     mov ah,9
0012     int 21h
0013
0014     |
0015
0016     mov dx,10
0017     mov ah,2
0018     int 21h
0019     mov dx,13
0020     mov ah,2
0021     int 21h
0022
0023     mov ah,01h
0024     int 21h
0025
0026     mov ch,0Ah
0027     mov cl,00h
0028
0029
0030     cmp al,3ah
0031     sub al,30h
0032     mov bh,al
0033     mov bl,01h
0034
0035     L1:
0036     mov dl,0dh
0037     mov ah,02h
0038     int 21h
0039
0040     mov dl,0ah
0041     mov ah,02h
0042     int 21h
0043
0044     mov dl,'0'
0045     mov ah,02h
0046     int 21h
0047
0048     mov dl,bh
0049     add dl,30h
0050     mov ah,02h
0051     int 21h
0052
0053     mov dl,'*'
0054     mov ah,02h
0055     int 21h
0056
0057     mov al,bl
0058     mul bh
0059
0060     AAM
0061
0062     mov

```

```

062 PUSH ax
063
064 mov ah,00h
065 mov al,bl
066
067 AAA
068
069 mov cl,ah
070 mov bl,al
071
072 mov dl,cl
073 add dl,30h
074 mov ah,02h
075 int 21h
076
077 mov dl,bl
078 add dl,30h
079 mov ah,02h
080 int 21h
081
082 mov dl,'='
083 mov ah,02h
084 int 21h
085
086 POP ax
087
088 mov dh,al
089 mov dl,ah
090 add dl,30h
091 mov ah,02h
092 int 21h
093
094 mov dl,dh
095 add dl,30h
096 mov ah,02h
097 int 21h
098
099
100
101 inc bl
102 dec ch
103 cmp ch,00h
104
105
106
107 jne L1
108
109
110 nov ah,4ch
111 int 21h
112
113
114
115 main endp
116 end main
117 ret
118
119
120

```



Q3: Write a program that loops through a string and pushes each character on the stack then POP the letter from the stack (in reverse order) and store them back into the same string variable. Because the stack is a LIFO structure, the letters in the string are reversed.

```
.model small
.stack 100h
.data

.code
main proc
    mov ax,05
    mov bx,10
    mov cx,15
    mov dx,20

    push ax
    push bx
    push cx
    push dx

    mov ax,0
    mov bx,0
    mov cx,0
    mov dx,0

    POP DX
    POP CX
    POP BX
    POP AX

main endp
end main

ret
```

stack	
0710:012A	9090
0710:0128	9090
0710:0126	9090
0710:0124	9090
0710:0122	9090
0710:0120	9090
0710:011E	585B
0710:011C	595A
0710:011A	0000
0710:0118	BA00
0710:0116	00B9
0710:0114	0000
0710:0112	BB00
0710:0110	00B8
0710:010E	5251
0710:010C	5350
0710:010A	0014
0710:0108	BA00
0710:0106	0FB9
0710:0104	000A
0710:0102	BB00
0710:0100	05B8

emulator: noname.exe

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	00	00
BX	00	00
CX	01	20
DX	00	00
CS	0720	

0720:0000 0720:0000

Address	Hex	ASCII	Comment
07200: B8 184	B8 184		
07201: 05 005	05 005		
07202: 00 000	00 000	NULL	
07203: B8 187	B8 187		
07204: 0A 010	0A 010	NEWL	
07205: 00 000	00 000	NULL	
07206: B9 185	B9 185		
07207: 0F 015	0F 015	*	
07208: 00 000	00 000	NULL	
07209: BA 186	BA 186		
0720A: 14 020	14 020		
0720B: 00 000	00 000	NULL	
0720C: 50 000	50 000	P	
0720D: 53 003	53 003	S	
0720E: 51 001	51 001	Q	
0720F: 52 002	52 002	R	
07210: B8 184	B8 184		
07211: 00 000	00 000	NULL	
07212: 00 000	00 000	NULL	

MOV AX, 00005h
MOV BX, 0000Ah
MOV CX, 0000Fh
MOV DX, 00014h
PUSH AX
PUSH BX
PUSH CX
PUSH DX
MOV AX, 00000h
MOV BX, 00000h
MOV CX, 00000h
MOV DX, 00000h
POP DX
POP CX
POP BX
POP AX
NOP
NOP
...

screen source reset aux vars debug stack flags