Computer Organization and Assembly Language Lab

LAB # 01

Submitted By
QASIM ALI (20P-0070)
Submitted to :KHURMA SHAHZAD

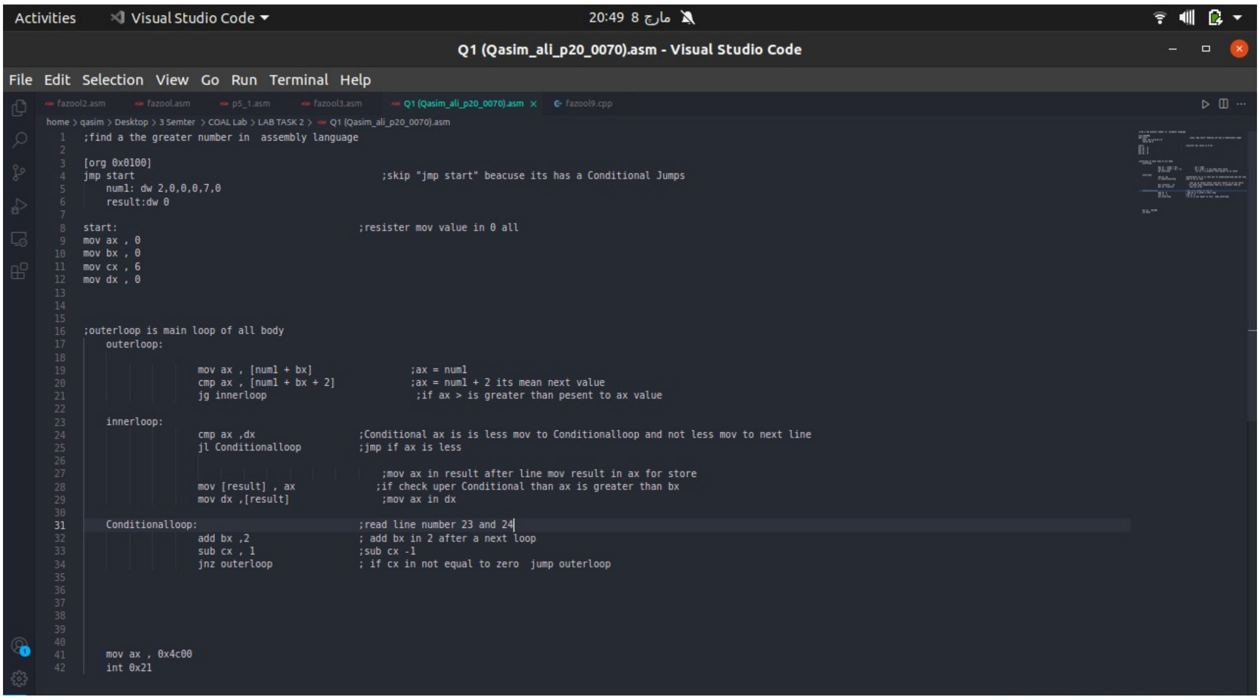
(INSTRUCTOR CS)

DEPARTMENT OF COMPUTER SCIENCE

FAST NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES, PESHAWAR

Session 2020-2024

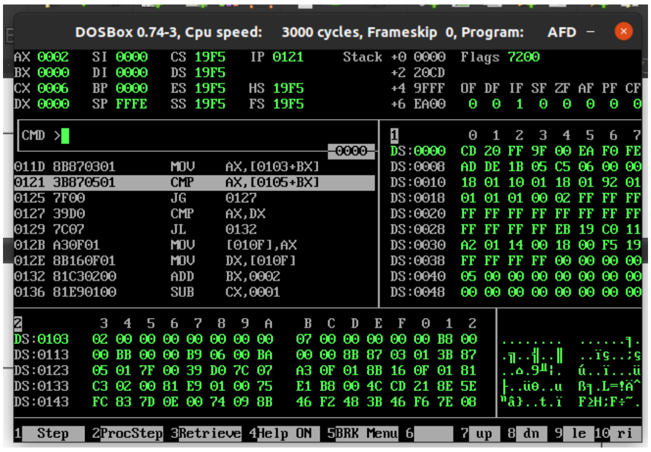
```
;Write an assembly language program to find largest digit of your Roll No and store it in
the memory.


[org 0x0100]
jmp start                                    ;skip "jmp start" beacuse its has a Conditional Jumps
   num1: dw 2,0,0,0,7,0
   result:dw 0

start:                                ;resister mov value in 0 all
mov ax , 0
mov bx , 0
mov cx , 6
mov dx , 0




;outerloop is main loop of all body
   outerloop:

             mov ax , [num1 + bx]             ;ax = num1
             cmp ax , [num1 + bx + 2]           ;ax = num1 + 2 its mean next value
             jg innerloop                    ;if ax > is greater than pesent to ax value

   innerloop:
             cmp ax ,dx             ;Conditional ax is is less mov to Conditionalloop and not less
mov to next line
             jl Conditionalloop          ;jmp if ax is less

                                  ;mov ax in result after line mov result in ax for store
             mov [result] , ax            ;if check uper Conditional than ax is greater than bx
             mov dx ,[result]              ;mov ax in dx


   Conditionalloop:                   ;read line number 25 and 26
             add bx ,2               ; add bx in 2 after a next loop
             sub cx , 1              ;sub cx -1
             jnz outerloop             ; if cx in not equal to zero  jump outerloop


   mov ax , 0x4c00
   int 0x21


Note: the greater number is save in dx resister.
```


Activities
Visual Studio Code
20:49 8 مارج
Q1 (Qasim_ali_p20_0070).asm - Visual Studio Code
File Edit Selection View Go Run Terminal Help
fazool2.asm
fazool.asm
p5_1.asm
fazool3.asm
Q1 (Qasim_ali_p20_0070).asm
fazool9.cpp
home > qasim > Desktop > 3 Semter > COAL Lab > LAB TASK 2 > Q1 (Qasim_ali_p20_0070).asm

```
1   ;find a the greater number in  assembly language
2
3   [org 0x0100]
4   jmp start                                           ;skip "jmp start" beacuse its has a Conditional Jumps
5       num1: dw 2,0,0,0,7,0
6       result:dw 0
7
8   start:                                          ;resister mov value in 0 all
9   mov ax , 0
10  mov bx , 0
11  mov cx , 6
12  mov dx , 0
13
14
15
16  ;outerloop is main loop of all body
17      outerloop:
18
19                  mov ax , [num1 + bx]                ;ax = num1
20                  cmp ax , [num1 + bx + 2]            ;ax = num1 + 2 its mean next value
21                  jg innerloop                         ;if ax > is greater than pesent to ax value
22
23      innerloop:
24                  cmp ax ,dx                      ;Conditional ax is is less mov to Conditionalloop and not less mov to next line
25                  jl Conditionalloop              ;jmp if ax is less
26
27                                                      ;mov ax in result after line mov result in ax for store
28                  mov [result] , ax                  ;if check uper Conditional than ax is greater than bx
29                  mov dx ,[result]                    ;mov ax in dx
30
31      Conditionalloop:                            ;read line number 23 and 24
32                  add bx ,2                       ; add bx in 2 after a next loop
33                  sub cx , 1                      ;sub cx -1
34                  jnz outerloop                   ; if cx in not equal to zero  jump outerloop
35
36
37
38
39
40
41      mov ax , 0x4c00
42      int 0x21
```

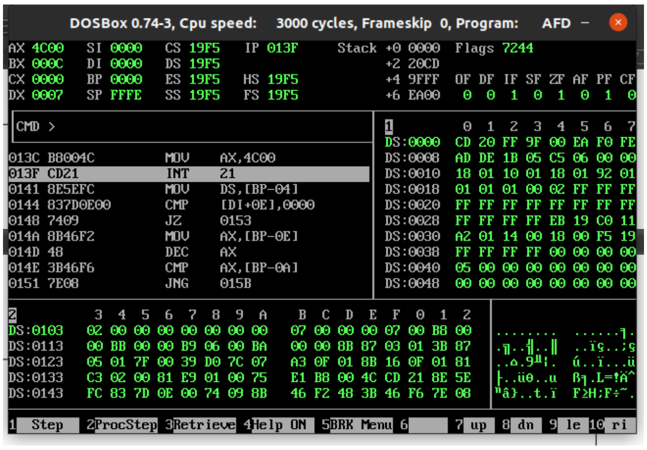
<u>**Screenshot**</u>:-

Screenshot of AFD debugger at the start of program.


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

```
AX 0002   SI 0000   CS 19F5   IP 0121     Stack +0 0000  Flags 7200
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0006   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 >                                     1        0  1  2  3  4  5  6  7
                                 -0000-    DS:0000  CD 20 FF 9F 00 EA F0 FE
011D 8B870301      MOV   AX,[0103+BX]      DS:0008  AD DE 1B 05 C5 06 00 00
0121 3B870501      CMP   AX,[0105+BX]      DS:0010  18 01 10 01 18 01 92 01
0125 7F00          JG    0127              DS:0018  01 01 01 00 02 FF FF FF
0127 39D0          CMP   AX,DX             DS:0020  FF FF FF FF FF FF FF FF
0129 7C07          JL    0132              DS:0028  FF FF FF FF EB 19 C0 11
012B A30F01        MOV   [010F],AX         DS:0030  A2 01 14 00 18 00 F5 19
012E 8B160F01      MOV   DX,[010F]         DS:0038  FF FF FF FF 00 00 00 00
0132 81C30200      ADD   BX,0002           DS:0040  05 00 00 00 00 00 00 00
0136 81E90100      SUB   CX,0001           DS:0048  00 00 00 00 00 00 00 00

2          3  4  5  6  7  8  9  A   B  C  D  E  F  0  1  2
DS:0103   02 00 00 00 00 00 00 00  07 00 00 00 00 00 B8 00   ........  ......╕.
DS:0113   00 BB 00 00 B9 06 00 BA  00 00 8B 87 03 01 3B 87   .╗..╣..║  ..ïç..;ç
DS:0123   05 01 7F 00 39 D0 7C 07  A3 0F 01 8B 16 0F 01 81   ..⌂.9╨|.  ú..ï...ü
DS:0133   C3 02 00 81 E9 01 00 75  E1 B8 00 4C CD 21 8E 5E   ├..üΘ..u  ß╕.L═!Ä^
DS:0143   FC 83 7D 0E 00 74 09 8B  46 F2 48 3B 46 F6 7E 08   ⁿâ}..t.ï  F≥H;F÷~.

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

Screenshot of AFD debugger showing the final values.
Note: the greater number is save in dx resister.



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

```
AX 4C00   SI 0000   CS 19F5   IP 013F     Stack +0 0000  Flags 7244
BX 000C   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 0007   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  1  0  1  0

 CMD >                                          1          0  1  2  3  4  5  6  7
                                                DS:0000   CD 20 FF 9F 00 EA F0 FE
013C B8004C          MOV     AX,4C00            DS:0008   AD DE 1B 05 C5 06 00 00
013F CD21            INT     21                 DS:0010   18 01 10 01 18 01 92 01
0141 8E5EFC          MOV     DS,[BP-04]         DS:0018   01 01 01 00 02 FF FF FF
0144 837D0E00        CMP     [DI+0E],0000       DS:0020   FF FF FF FF FF FF FF FF
0148 7409            JZ      0153               DS:0028   FF FF FF FF EB 19 C0 11
014A 8B46F2          MOV     AX,[BP-0E]         DS:0030   A2 01 14 00 18 00 F5 19
014D 48              DEC     AX                 DS:0038   FF FF FF FF 00 00 00 00
014E 3B46F6          CMP     AX,[BP-0A]         DS:0040   05 00 00 00 00 00 00 00
0151 7E08            JNG     015B               DS:0048   00 00 00 00 00 00 00 00

2          3  4  5  6  7  8  9  A    B  C  D  E  F  0  1  2
DS:0103   02 00 00 00 00 00 00 00   07 00 00 00 07 00 B8 00   ........  ......╕.
DS:0113   00 BB 00 00 B9 06 00 BA   00 00 8B 87 03 01 3B 87   .╗..╣..║  ..ïç..;ç
DS:0123   05 01 7F 00 39 D0 7C 07   A3 0F 01 8B 16 0F 01 81   ..⌂.9╨|.  ú..ï...ü
DS:0133   C3 02 00 81 E9 01 00 75   E1 B8 00 4C CD 21 8E 5E   ├..üΘ..u  ß╕.L═!Ä^
DS:0143   FC 83 7D 0E 00 74 09 8B   46 F2 48 3B 46 F6 7E 08   ⁿâ}..t.ï  F≥H;F÷~.

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