

# LAB 2

Name : Dhairy Agrawal

ID: 2018B4A70827G

## TASK 1

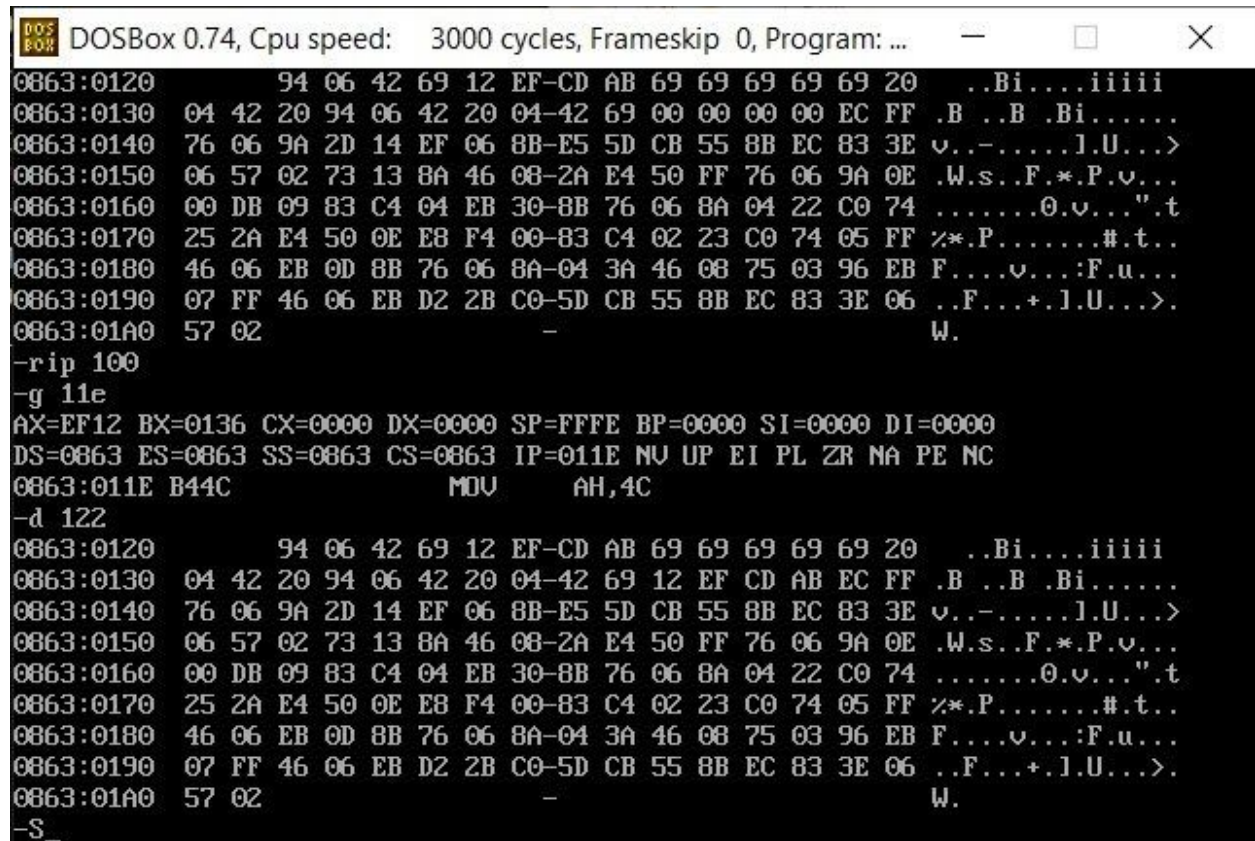
- Write an ALP that finds the maximum number from a set of 32-bit numbers

### Code:

```
.model tiny
.486
.data
arr dd 69420694h,0abcdef12h,69696969h,42042069h,42069420h,69420420h
res dd 0
.code
.startup
    lea bx,arr          //stores the offset
    mov eax,[bx]        //loads the data stored into eax at the address stored in
bx
    mov cl,5h           //cl is used to count here we will use 5 comps for 6 nums
x1:  inc bx              //inc bx by 1
    add bx,2h           //add bx by 2
    inc bx              //inc bx by 1
    cmp eax,[bx]        //compare numbers
    jae x2              // if number is not greater than then move to x2
    mov eax,[bx]        //if number was bigger than move it in eax
x2:  dec cl              //decrement the count
    jnz x1              //if zero flag is not set then move to x1
    mov res, eax        //move eax to res
.exit
```

end

## Screen Shot-



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
0063:0120  94 06 42 69 12 EF-CD AB 69 69 69 69 20  ..Bi....iiii
0063:0130  04 42 20 94 06 42 20 04-42 69 00 00 00 EC FF .B .B .Bi.....
0063:0140  76 06 9A 2D 14 EF 06 8B-E5 5D CB 55 8B EC 83 3E v..-.....l.U...>
0063:0150  06 57 02 73 13 8A 46 08-2A E4 50 FF 76 06 9A 0E .W.s..F*.P.v...
0063:0160  00 DB 09 83 C4 04 EB 30-8B 76 06 8A 04 22 C0 74 .....0.v..."t
0063:0170  25 2A E4 50 0E E8 F4 00-83 C4 02 23 C0 74 05 FF %*.P.....#.t..
0063:0180  46 06 EB 0D 8B 76 06 8A-04 3A 46 08 75 03 96 EB F....v...:F.u...
0063:0190  07 FF 46 06 EB D2 2B C0-5D CB 55 8B EC 83 3E 06 ..F...+.l.U...>.
0063:01A0  57 02                -                W.
-rip 100
-g 11e
AX=EF12 BX=0136 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0063 ES=0063 SS=0063 CS=0063 IP=011E NU UP EI PL ZR NA PE NC
0063:011E B44C                MOV     AH,4C
-d 122
0063:0120  94 06 42 69 12 EF-CD AB 69 69 69 69 20  ..Bi....iiii
0063:0130  04 42 20 94 06 42 20 04-42 69 12 EF CD AB EC FF .B .B .Bi.....
0063:0140  76 06 9A 2D 14 EF 06 8B-E5 5D CB 55 8B EC 83 3E v..-.....l.U...>
0063:0150  06 57 02 73 13 8A 46 08-2A E4 50 FF 76 06 9A 0E .W.s..F*.P.v...
0063:0160  00 DB 09 83 C4 04 EB 30-8B 76 06 8A 04 22 C0 74 .....0.v..."t
0063:0170  25 2A E4 50 0E E8 F4 00-83 C4 02 23 C0 74 05 FF %*.P.....#.t..
0063:0180  46 06 EB 0D 8B 76 06 8A-04 3A 46 08 75 03 96 EB F....v...:F.u...
0063:0190  07 FF 46 06 EB D2 2B C0-5D CB 55 8B EC 83 3E 06 ..F...+.l.U...>.
0063:01A0  57 02                -                W.
-S_
```

## Task 2

- Write an ALP to add 2 16-byte nos. using them

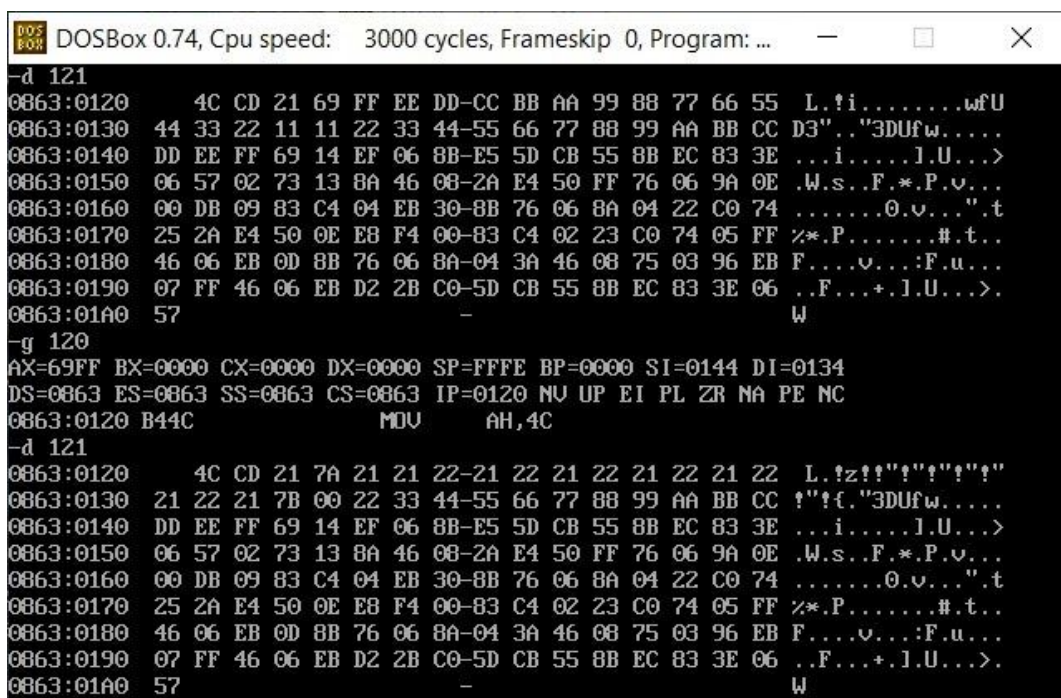
### Part A:

a. as 16-bit data

## CODE:-

```
.model tiny
.486
.data
dat1 db 69h,0ffh,0eeh,0ddh,0cch,0bbh,0aah,99h,88h,77h,66h,55h,44h,33h,22h,11h
dat2 db 11h,22h,33h,44h,55h,66h,77h,88h,99h,0aah,0bbh,0cch,0ddh,0eeh,0ffh,69h
.code
.startup
    lea di,dat1      //stores the starting offset
    lea si,dat2      //stores the starting offset
    mov cl,8h        //stores the count of operations performed
    mov bl,0h        //stores the carry
x1: mov ax,[si]      //move 2 bytes(16bits) of data into ax
    adc [di],ax      //add with carry the 2 byts(16bits)
    add si,2h        //increment si by 2
    add di,2h        //increment di by 2
    dec cl           //decrement the count by 1
    jnz x1           //check zero flag
    jnc x2           //check carry flag
    inc bl           //increment the count
x2: mov [di],bl      //move to di address of bl
.exit
end
```

## ScreenShot



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
-d 121
0063:0120 4C CD 21 69 FF EE DD-CC BB AA 99 88 77 66 55 L.?i.....wfU
0063:0130 44 33 22 11 11 22 33 44-55 66 77 88 99 AA BB CC D3"...3DUfw....
0063:0140 DD EE FF 69 14 EF 06 8B-E5 5D CB 55 8B EC 83 3E ...i.....l.U...>
0063:0150 06 57 02 73 13 8A 46 08-2A E4 50 FF 76 06 9A 0E .W.s..F.*.P.v...
0063:0160 00 DB 09 83 C4 04 EB 30-8B 76 06 8A 04 22 C0 74 .....0.v..."t
0063:0170 25 2A E4 50 0E E8 F4 00-83 C4 02 23 C0 74 05 FF %*.P.....#.t..
0063:0180 46 06 EB 0D 8B 76 06 8A-04 3A 46 08 75 03 96 EB F....v...:F.u...
0063:0190 07 FF 46 06 EB D2 2B C0-5D CB 55 8B EC 83 3E 06 ..F...+.l.U...>.
0063:01A0 57 - W
-g 120
AX=69FF BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0144 DI=0134
DS=0063 ES=0063 SS=0063 CS=0063 IP=0120 NU UP EI PL ZR NA PE NC
0063:0120 B44C MOV AH,4C
-d 121
0063:0120 4C CD 21 7A 21 21 22-21 22 21 22 21 22 21 22 L.?z!?!?!?!?!
0063:0130 21 22 21 7B 00 22 33 44-55 66 77 88 99 AA BB CC ?"!{."3DUfw....
0063:0140 DD EE FF 69 14 EF 06 8B-E5 5D CB 55 8B EC 83 3E ...i.....l.U...>
0063:0150 06 57 02 73 13 8A 46 08-2A E4 50 FF 76 06 9A 0E .W.s..F.*.P.v...
0063:0160 00 DB 09 83 C4 04 EB 30-8B 76 06 8A 04 22 C0 74 .....0.v..."t
0063:0170 25 2A E4 50 0E E8 F4 00-83 C4 02 23 C0 74 05 FF %*.P.....#.t..
0063:0180 46 06 EB 0D 8B 76 06 8A-04 3A 46 08 75 03 96 EB F....v...:F.u...
0063:0190 07 FF 46 06 EB D2 2B C0-5D CB 55 8B EC 83 3E 06 ..F...+.l.U...>.
0063:01A0 57 - W
```

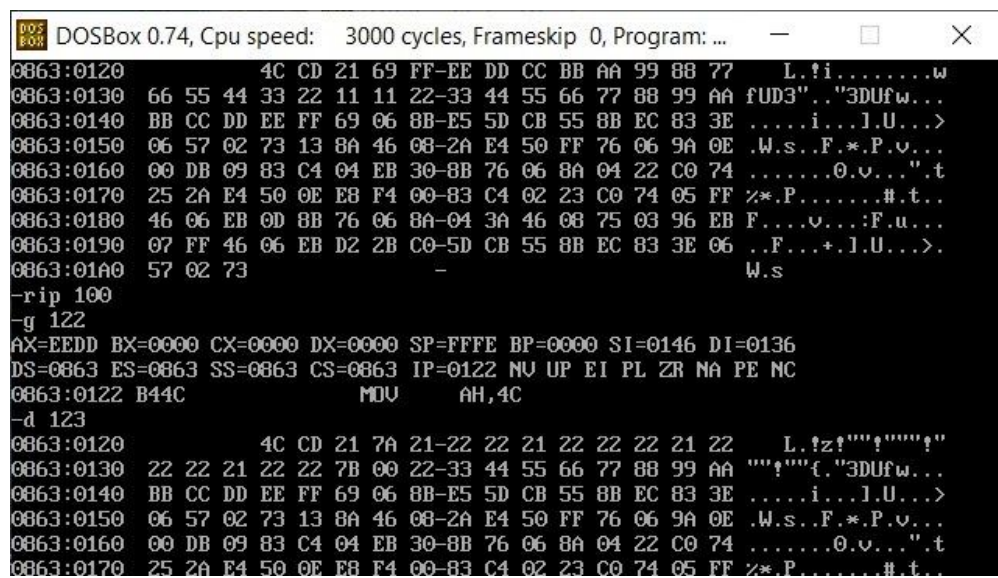
## Part B:

*as 32-bit data*

## CODE:-

```
.model tiny
.486
.data
dat1 db 69h,0ffh,0eeh,0ddh,0cch,0bbh,0aah,99h,88h,77h,66h,55h,44h,33h,22h,11h
dat2 db 11h,22h,33h,44h,55h,66h,77h,88h,99h,0aah,0bbh,0cch,0ddh,0eeh,0ffh,69h
.code
.startup
    lea di,dat1      //stores the starting offset
    lea si,dat2      //stores the starting offset
    mov cl,4h        //stores the count of operations performed
    mov bl,0h        //stores the carry
x1: mov eax,[si]      //move 4 bytes(32bits) of data into eax
    adc [di],eax      //add with carry the 4 bytes(32bits)
    add si,4h        //increment si by 4
    add di,4h        //increment di by 2
    dec cl           //decrement the count by 1
    jnz x1           //check zero flag
    jnc x2
    inc bl           //increment the count
x2: mov [di],bl       //move to di address of bl
.exit
end
```

## ScreenShot



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
0063:0120 4C CD 21 69 FF-EE DD CC BB AA 99 88 77 L.fti.....w
0063:0130 66 55 44 33 22 11 11 22-33 44 55 66 77 88 99 AA fUD3"..3DUfw...
0063:0140 BB CC DD EE FF 69 06 8B-E5 5D CB 55 8B EC 83 3E ....i...l.U...>
0063:0150 06 57 02 73 13 8A 46 08-2A E4 50 FF 76 06 9A 0E .W.s..F.*.P.v...
0063:0160 00 DB 09 83 C4 04 EB 30-8B 76 06 8A 04 22 C0 74 .....0.v..."t
0063:0170 25 2A E4 50 0E EB F4 00-83 C4 02 23 C0 74 05 FF %*.P.....#.t..
0063:0180 46 06 EB 0D 8B 76 06 8A-04 3A 46 08 75 03 96 EB F....v...:F.u...
0063:0190 07 FF 46 06 EB D2 2B C0-5D CB 55 8B EC 83 3E 06 ..F...+.l.U...>.
0063:01A0 57 02 73 - W.s
-rip 100
-g 122
AX=EEDD BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0146 DI=0136
DS=0863 ES=0863 SS=0863 CS=0863 IP=0122 NU UP EI PL ZR NA PE NC
0063:0122 B44C MOV AH,4C
-d 123
0063:0120 4C CD 21 7A 21-22 22 21 22 22 22 21 22 L.ftz!""t""t""t"
0063:0130 22 22 21 22 22 7B 00 22-33 44 55 66 77 88 99 AA ""!""t..3DUfw...
0063:0140 BB CC DD EE FF 69 06 8B-E5 5D CB 55 8B EC 83 3E ....i...l.U...>
0063:0150 06 57 02 73 13 8A 46 08-2A E4 50 FF 76 06 9A 0E .W.s..F.*.P.v...
0063:0160 00 DB 09 83 C4 04 EB 30-8B 76 06 8A 04 22 C0 74 .....0.v..."t
0063:0170 25 2A E4 50 0E EB F4 00-83 C4 02 23 C0 74 05 FF %*.P.....#.t..
```