LAB 2

Name : Dhairy Agrawal ID: 2018B4A70827G

TASK 1

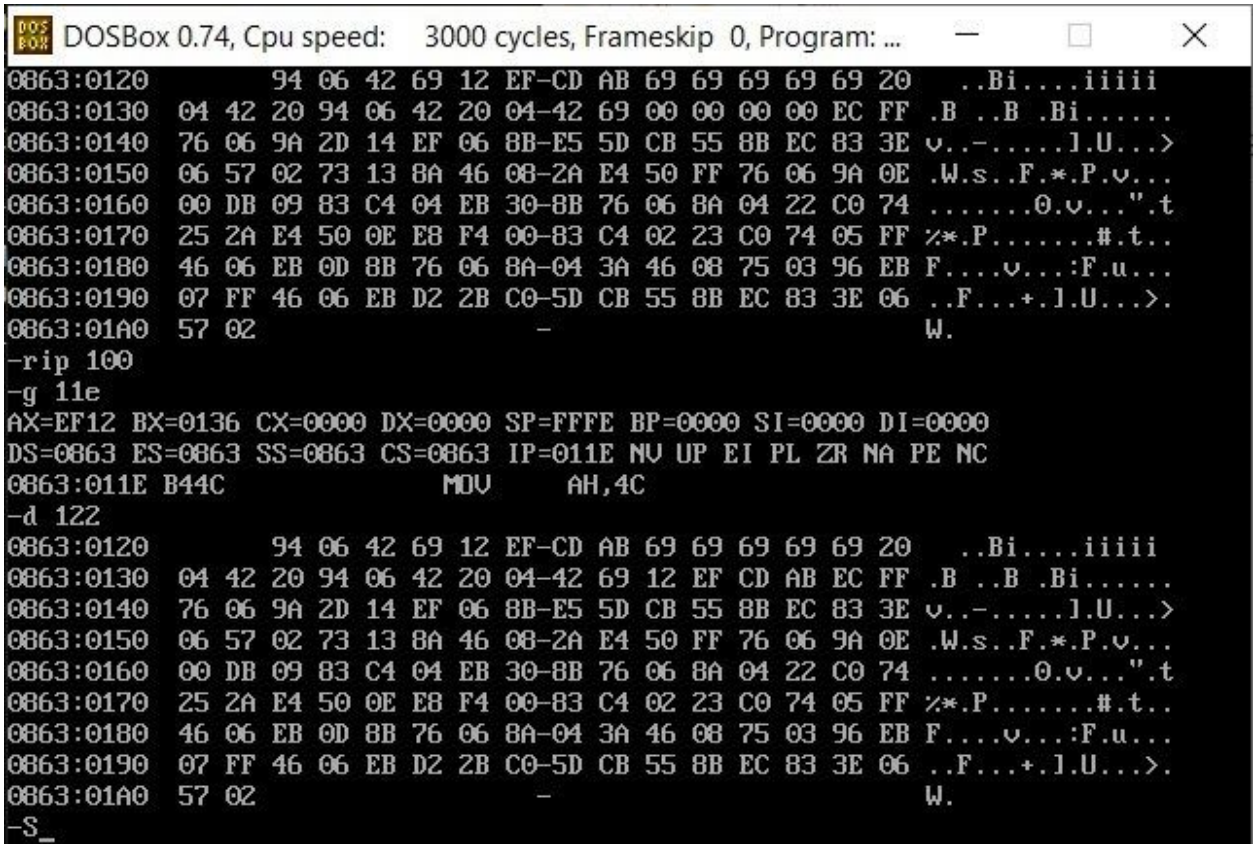
- *Write an ALP that finds the maximum number from a set of 32-bit numbers*

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-



Task 2

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

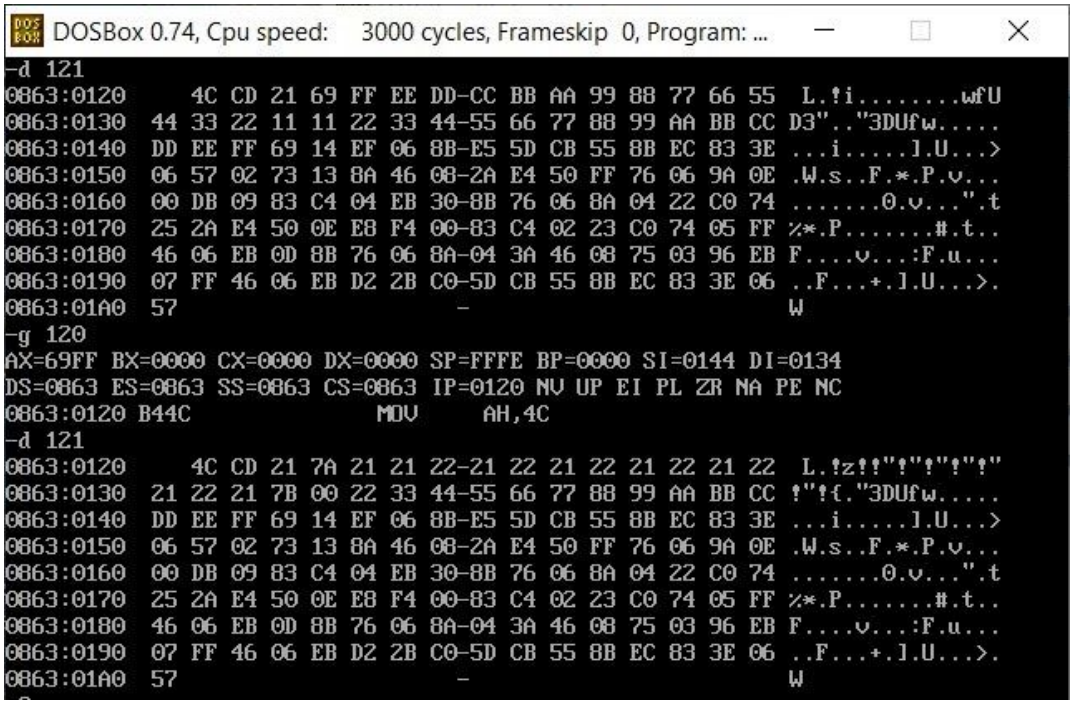
Part A:

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
   inc bl               //increment the count
x2: mov [di],bl         //move to di address of bl
.exit
end
```

ScreenShot



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