CSE2006-MICROPROCESSOR AND INTERFACING
LAB EXPERIMENT

NAME: OM ASHISH MISHRA

REGISTRATION NUMBER: 16BCE0789

SLOT: B2

1. To find the factorial of a number using MASM

Write an ALP
1. To find the factorial of a number using MASM

Aim: Factorial of a number

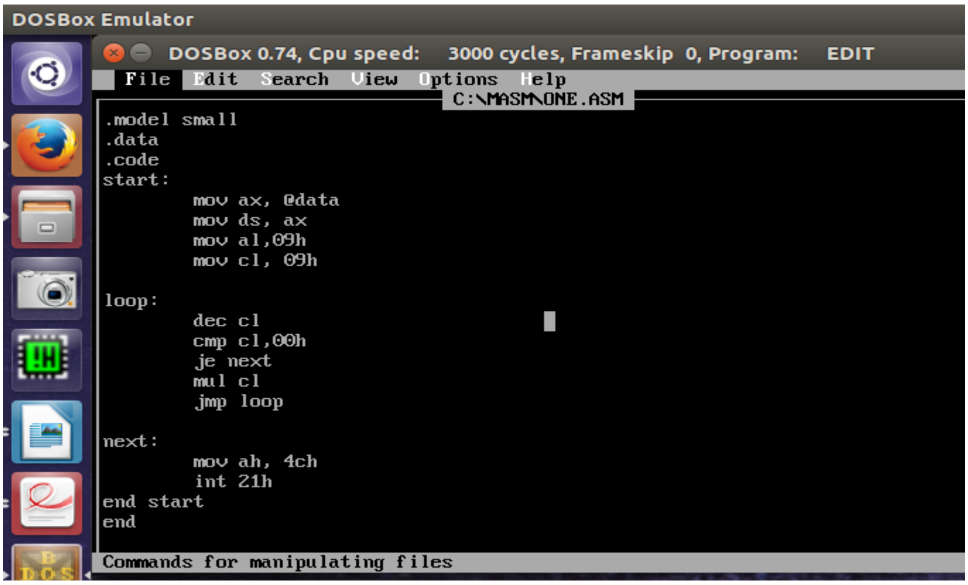
Algorithm:
1: First we take the number and store it in data segement
2: Then we run a loop to multiply it to a counter and decrease the value of the counter after every multiplication.
3. Then we print the result.

Sample Input

Input: 9
Sample Output

output: 362880

Result:
The result is 362880.


DOSBox Emulator
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
C:\MASM\ONE.ASM
Commands for manipulating files

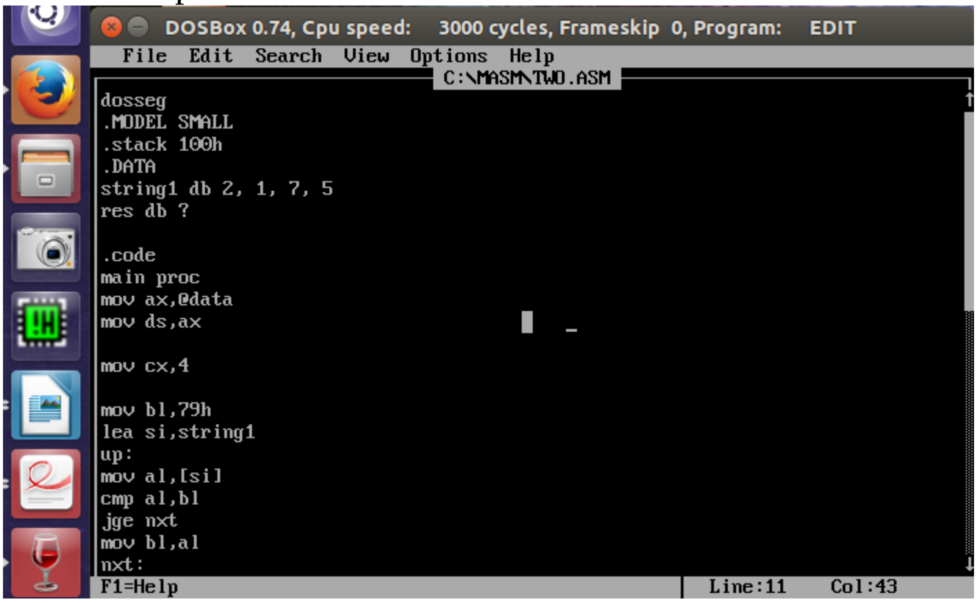
```
.model small
.data
.code
start:
        mov ax, @data
        mov ds, ax
        mov al,09h
        mov cl, 09h

loop:
        dec cl
        cmp cl,00h
        je next
        mul cl
        jmp loop

next:
        mov ah, 4ch
        int 21h
end start
end
```

2. To find the smallest element in an array using Emu 8086 /MASM

Aim: smallest element in an array
Algorithm:
1: First we take the numbers in an array
2: Then we use the logic of the bubble sort in order to sort the elements

    1. after taking the input we have to keep the value of counter in count and data in ax
    2. we store the value of the element in al and comapre with the next element
    3. if the current element is greater then we swap elements otherwise we leave them as it is.

3. Then we print the smallest element


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
C:\MASM\TWO.ASM
F1=Help
Line:11 Col:43

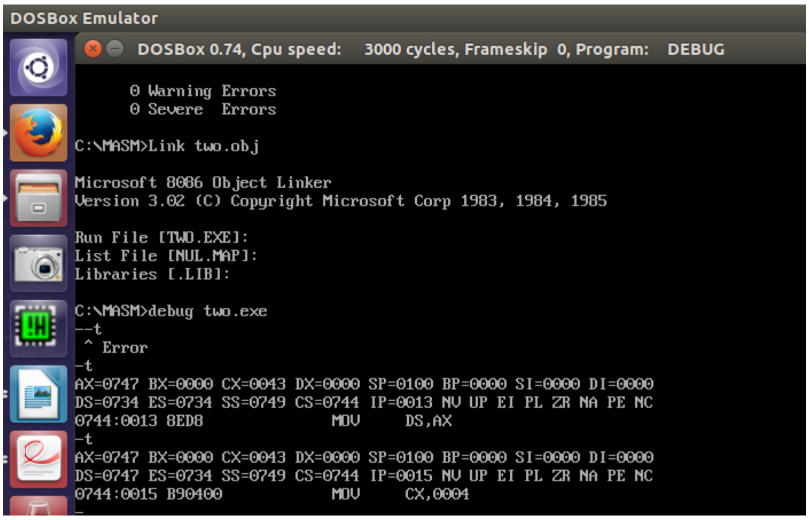
```
dosseg
.MODEL SMALL
.stack 100h
.DATA
string1 db 2, 1, 7, 5
res db ?

.code
main proc
mov ax,@data
mov ds,ax

mov cx,4

mov bl,79h
lea si,string1
up:
mov al,[si]
cmp al,bl
jge nxt
mov bl,al
nxt:
```

Sample Input
Input: 2,1,5,7

Sample Output

output: 1

Result:
The result is 1.


DOSBox Emulator
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
       0 Warning Errors
       0 Severe  Errors

C:\MASM>Link two.obj

Microsoft 8086 Object Linker
Version 3.02 (C) Copyright Microsoft Corp 1983, 1984, 1985

Run File [TWO.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\MASM>debug two.exe
--t
 ^ Error
-t
AX=0747 BX=0000 CX=0043 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=0734 ES=0734 SS=0749 CS=0744 IP=0013 NV UP EI PL ZR NA PE NC
0744:0013 8ED8              MOV     DS,AX
-t
AX=0747 BX=0000 CX=0043 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=0747 ES=0734 SS=0749 CS=0744 IP=0015 NV UP EI PL ZR NA PE NC
0744:0015 B90400            MOV     CX,0004
-
```

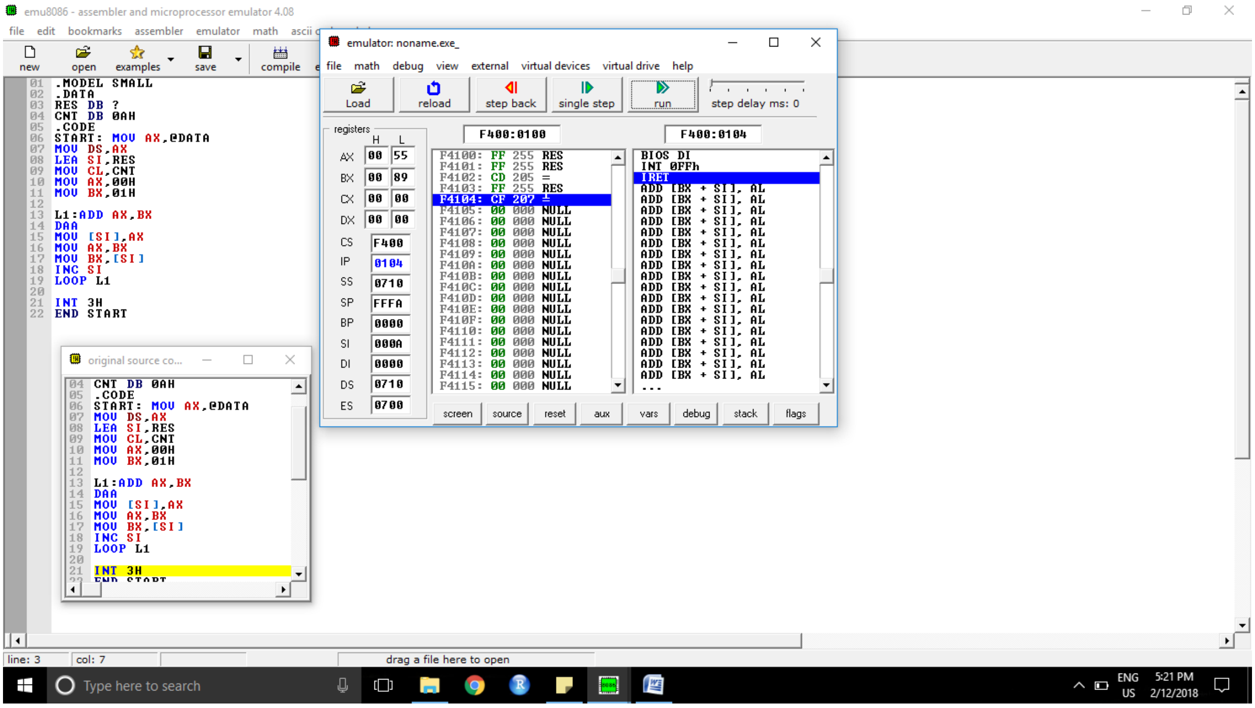
3. To generate the Fibonacci series for N terms.

ALP:

```
.MODEL SMALL
.DATA
RES DB ?
CNT DB 0AH
.CODE
START: MOV AX,@DATA
MOV DS,AX
LEA SI,RES
MOV CL,CNT
MOV AX,00H
MOV BX,01H
L1:ADD AX,BX
DAA
MOV [SI],AX
MOV AX,BX
MOV BX,[SI]
INC SI
LOOP L1
INT 3H
END START
```

Algorithm:

1. We take to number of times the Fibonacci series has to continue.
2. Then we initialize ax = 1 and bx=1 and res to store the result
3. After completing data segment we go for code segment
4. We run a loop where we add ax and bx
5. Put the value in source from ax
6. Put the value of ax in bx
7. Put the value of si into ax
8. Increase the value of si
9. Continue the loop
10. Thus terminate at the end and end the loop.



emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii
new open examples save compile
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
screen source reset aux vars debug stack flags
original source co...
line: 3 col: 7 drag a file here to open
Type here to search
ENG US 5:21 PM 2/12/2018

```
.MODEL SMALL
.DATA
RES DB ?
CNT DB 0AH
.CODE
START: MOV AX,@DATA
MOV DS,AX
LEA SI,RES
MOV CL,CNT
MOV AX,00H
MOV BX,01H

L1:ADD AX,BX
DAA
MOV [SI],AX
MOV AX,BX
MOV BX,[SI]
INC SI
LOOP L1

INT 3H
END START
```

```
      H   L
AX   00  55
BX   00  89
CX   00  00
DX   00  00
CS   F400
IP   0104
SS   0710
SP   FFFA
BP   0000
SI   000A
DI   0000
DS   0710
ES   0700
```

```
F400:0100

F4100: FF 255 RES
F4101: FF 255 RES
F4102: CD 205 =
F4103: FF 255 RES
F4104: CF 207 ±
F4105: 00 000 NULL
F4106: 00 000 NULL
F4107: 00 000 NULL
F4108: 00 000 NULL
F4109: 00 000 NULL
F410A: 00 000 NULL
F410B: 00 000 NULL
F410C: 00 000 NULL
F410D: 00 000 NULL
F410E: 00 000 NULL
F410F: 00 000 NULL
F4110: 00 000 NULL
F4111: 00 000 NULL
F4112: 00 000 NULL
F4113: 00 000 NULL
F4114: 00 000 NULL
F4115: 00 000 NULL
```

```
F400:0104

BIOS DI
INT 0FFh
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

```
CNT DB 0AH
.CODE
START: MOV AX,@DATA
MOV DS,AX
LEA SI,RES
MOV CL,CNT
MOV AX,00H
MOV BX,01H

L1:ADD AX,BX
DAA
MOV [SI],AX
MOV AX,BX
MOV BX,[SI]
INC SI
LOOP L1

INT 3H
END START
```

Input Sample:

10 is the number of times

Output Sample:

89 is the output.

Result:

The output of the Fibonacci series is 89h.