1)

%macro print 2

mov eax,4

mov ebx,1

mov ecx,%1

mov edx,%2

int 80h

%endmacro

input:

mov eax,3

mov ebx,2

mov ecx,n

mov edx,9

int 80h

retṇ

section .bss

n resd 9

a resb 9

b resb 9

cnt resb 9

section .text

fibonaaci:

print msg1,len1

call input

print msg2,len2

mov al,'0'

mov [a],al

mov al,'1'

mov [b],al

cmp byte[n],'0' ;N=0

jle exit

print a,9

cmp byte[n],'2';N>=1

jl exit

print b,9

cmp byte[n],'3' ;N>=2

jl exit

mov al,'2' ;LOOP COUNTER INITIALIZED TO 2

mov [cnt],al

fibo:

mov eax,[a]

sub eax,'0'

mov ebx,[b]

sub ebx,'0'

add eax,ebx

add eax,'0'

add ebx,'0'

mov [a],ebx

mov [b],eax

print b,9

inc byte[cnt]

mov al,[cnt]

cmp byte[n],al

je exit

jmp fibo

ret

section .data

msg1 db 'ENTER THE VALUE OF N (1-9): '

len1 equ $-msg1

msg2 db 10,'FIBONACCI SERIES: '

len2 equ $-msg2

nextline db' ',10

len3 equ $-nextline

section .text

global \_start:

\_start:

call fibonaaci

exit:

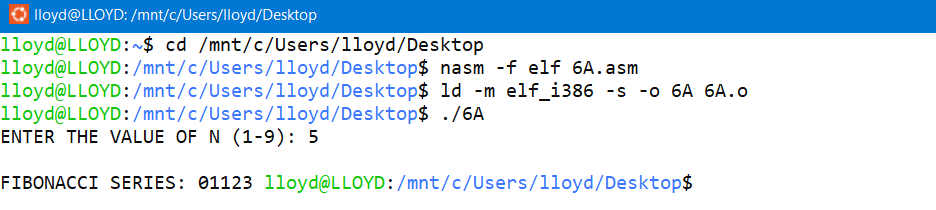
print nextline,1

mov eax,1

mov ebx,0

int 80h

OUTPUT



2)

input:

mov eax,3

mov ebx,2

ret

print:

mov eax,4

mov ebx,1

ret

factorial:

mov al,'1'

mov cl,'1'

fact:

cmp cl,byte[n]

jg display

sub al,'0' ;Multiplication

mov bl,cl

sub bl,'0'

mul bl

add al,'0'

inc cl

jmp fact

display:

mov [n],al

call print

mov ecx,n

mov edx,8

int 80h

ret

section .data

msg1 db 'ENTER A NUMBER: '

len1 equ $-msg1

msg2 db 10,'THE FACTORIAL: '

len2 equ $-msg2

section .bss

n resb 9

section .text

global \_start:

\_start:

call print

mov ecx,msg1

mov edx,len1

int 80h

call input

mov ecx,n

mov edx,8

int 80h

call print

mov ecx,msg2

mov edx,len2

int 80h

call factorial

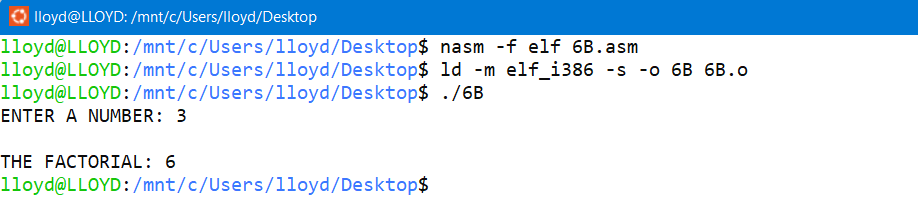
exit:

mov eax,1

mov ebx,0

int 80h

OUTPUT



3)

addition:

mov eax,[num1]

sub eax,'0'

mov ebx,[num2]

sub ebx,'0'

add eax,ebx

add eax, '0'

mov[result],eax

mov eax, sys\_out

mov ebx, stdout

mov ecx, p3

mov edx, p3L

int 80h

mov eax, sys\_out

mov ebx, stdout

mov ecx, result

mov edx, 9

int 80h

ret

substraction:

mov eax,[num1]

sub eax,'0'

mov ebx,[num2]

sub ebx,'0'

sub eax,ebx

add eax, '0'

mov[result],eax

mov eax, sys\_out

mov ebx, stdout

mov ecx, p4

mov edx, p4L

int 80h

mov eax, sys\_out

mov ebx, stdout

mov ecx, result

mov edx, 9

int 80h

ret

multiply:

mov al,[num1]

sub al,'0'

mov bl,[num2]

sub bl,'0'

mul bl

add ax, '0'

mov[result],ax

mov eax, sys\_out

mov ebx, stdout

mov ecx, p5

mov edx, p5L

int 80h

mov eax, sys\_out

mov ebx, stdout

mov ecx, result

mov edx, 9

int 80h

ret

division:

mov al,[num1]

sub al,'0'

mov bl,[num2]

sub bl,'0'

div bl

add ah, '0'

add al, '0'

mov[result1],ah

mov[result2],al

mov eax, sys\_out

mov ebx, stdout

mov ecx, p6

mov edx, p6L

int 80h

mov eax, sys\_out

mov ebx, stdout

mov ecx, result1

mov edx, 9

int 80h

mov eax, sys\_out

mov ebx, stdout

mov ecx, p7

mov edx, p7L

int 80h

mov eax, sys\_out

mov ebx, stdout

mov ecx, result2

mov edx, 9

int 80h

ret

num\_input:

mov eax, sys\_out

mov ebx, stdout

mov ecx, p1

mov edx, p1L

int 80h

mov eax, sys\_in

mov ebx, stdin

mov ecx, num1

mov edx, 9

int 80h

mov eax, sys\_out

mov ebx, stdout

mov ecx, p2

mov edx, p2L

int 80h

mov eax, sys\_in

mov ebx, stdin

mov ecx, num2

mov edx, 9

int 80h

ret

section .bss

num1 resb 9

num2 resb 9

result resb 9

result1 resb 9

result2 resb 9

section .data

sys\_out equ 4 ;To output

sys\_in equ 3 ;To Input

stdout equ 1 ;Stdout

stdin equ 2 ;Stdins

p1 db 'ENTER THE FIRST NUMBER: '

p1L equ $-p1

p2 db 'ENTER THE SECOND NUMBER: '

p2L equ $-p2

p3 db 'ADD= '

p3L equ $-p3

p4 db 10,'SUB= '

p4L equ $-p4

p5 db 10,'MUL= '

p5L equ $-p5

p6 db 10,'R= '

p6L equ $-p6

p7 db 9,'Q= '

p7L equ $-p7

nextline db ' ',10

nl equ $-nextline

section .text

global \_start:

\_start:

call num\_input

call addition

call substraction

call multiply

call division

mov eax,1

mov ebx,0

int 80h

OUTPUT

