1)

%macro print 2

mov eax,4

mov ebx,1

mov ecx,%1

mov edx,%2

int 80h

%endmacro

section .data

num1 db '2022'

section .text

global \_start:

\_start:

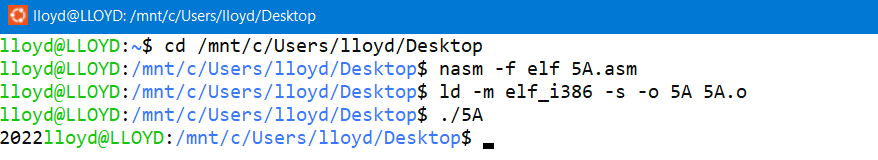
print num1,4

mov eax,1

mov ebx,0

int 80h

OUTPUT



2)

%macro write 2

mov eax,4

mov ebx,1

mov ecx,%1

mov edx,%2

int 80h

%endmacro

section .data

mss1 db 'ENTER FIRST NUMBER: '

len1 equ $-mss1

mss2 db 'ENTER SECOND NUMBER: '

len2 equ $-mss2

mss3 db 'THE NUMBERS YOU ENTERED ARE: ',10

len3 equ $-mss3

section .bss

num1 resb 9

num2 resb 9

section .text

global \_start:

\_start:

write mss1,len1

mov eax,3

mov ebx,2

mov ecx,num1

mov edx,9

int 80h

write mss2,len2

mov eax,3

mov ebx,2

mov ecx,num2

mov edx,9

int 80h

write mss3,len3

write num1,9

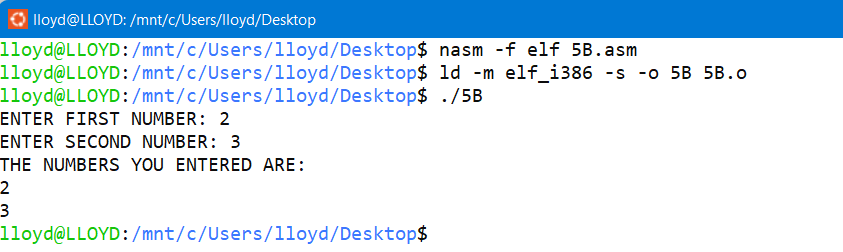
write num2,9

mov eax,1

mov ebx,0

int 80h

OUTPUT



3)

%macro read 2

mov eax,3

mov ebx,2

mov ecx,%1

mov edx,%2

int 80h

%endmacro

section .data

mss1 db 'ENTER FIRST NUMBER: '

len1 equ $-mss1

mss2 db 'THE NUMBER YOU ENTERED IS: '

len2 equ $-mss2

section .bss

num1 resb 9

section .text

global \_start:

\_start:

mov eax,4

mov ebx,1

mov ecx,mss1

mov edx,len1

int 80h

read num1,9

mov eax,4

mov ebx,1

mov ecx,mss2

mov edx,len2

int 80h

mov eax,4

mov ebx,1

mov ecx,num1

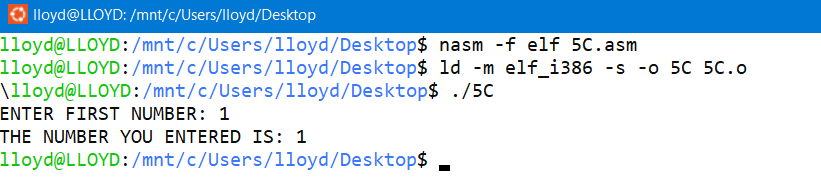
mov edx,9

int 80h

mov eax,1

mov ebx,0

int 80h

OUTPUT

4)

%macro read 4

mov eax,4 ;Input 1

mov ebx,1

mov ecx,mss1

mov edx,len1

int 80h

mov eax,3

mov ebx,2

mov ecx,%1

mov edx,%2

int 80h

mov eax,4 ;Input 2

mov ebx,1

mov ecx,mss2

mov edx,len2

int 80h

mov eax,3

mov ebx,2

mov ecx,%3

mov edx,%4

int 80h

%endmacro

section .data

mss1 db 'ENTER FIRST NUMBER: '

len1 equ $-mss1

mss2 db 'ENTER SECOND NUMBER: '

len2 equ $-mss2

mss3 db 'THE NUMBERS YOU ENTERED ARE',10

len3 equ $-mss3

section .bss

num1 resb 9

num2 resb 9

section .text

global \_start:

\_start:

read num1,9,num2,9

mov eax,4

mov ebx,1

mov ecx,mss3

mov edx,len3

int 80h

mov eax,4

mov ebx,1

mov ecx,num1

mov edx,9

int 80h

mov eax,4

mov ebx,1

mov ecx,num2

mov edx,9

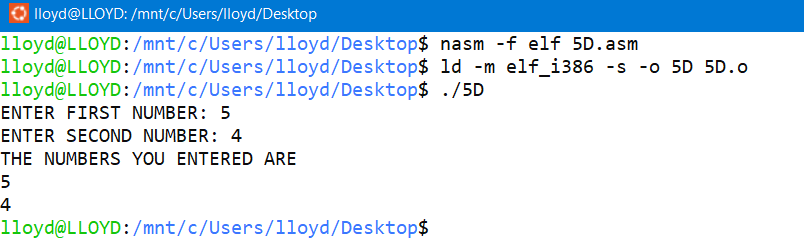
int 80h

mov eax,1

mov ebx,0

int 80h

OUTPUT



5)

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

%macro addition 0

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

%endmacro

%macro substraction 0

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

%endmacro

%macro multiply 0

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

%endmacro

%macro division 0

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

%endmacro

%macro num\_input 0

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

%endmacro

section .text

global \_start:

\_start:

num\_input

addition

substraction

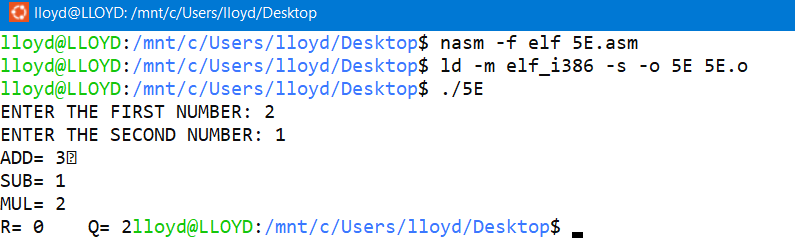
multiply

division

mov eax,1

mov ebx,0

int 80h

OUTPUT

6)

%macro print 2

mov eax,4

mov ebx,1

mov ecx,%1

mov edx,%2

int 80h

%endmacro

%macro input 2

mov eax,3

mov ebx,2

mov ecx,%1

mov edx,%2

int 80h

%endmacro

%macro fib 0

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

%endmacro

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

section .bss

n resd 9

a resb 9

b resb 9

cnt resb 9

section .text

global \_start:

\_start:

print msg1,len1

input n,9

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 ;N>=1

cmp byte[n],'2'

jl exit

print b,9 ;N>=2

cmp byte[n],'3'

jl exit

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

mov [cnt],al

fibo:

fib

je exit

jmp fibo

exit:

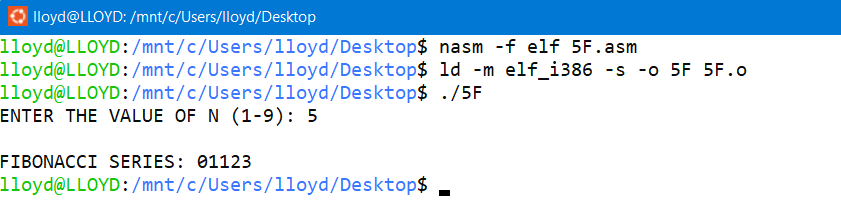
print nextline,1

mov eax,1

mov ebx,0

int 80h

OUTPUT



7)

%macro printer 2

mov eax,4

mov ebx,1

mov ecx,%1

mov edx,%2

int 80h

%endmacro

section .bss

count resw 9

section .data

my\_name db 'LLOYD ALRICH COSTA',10

len1 equ $-my\_name

section .text

global \_start:

\_start:

mov al,'0'

mov [count],al

loop:

printer my\_name,len1

inc word[count]

mov al,[count]

cmp al,'7'

jl loop

jg exit

exit:

mov eax,1

mov ebx,0

int 80h

OUTPUT

