//alp to print the factorial of a number using macros

%macro read 2

mov eax, 3

mov ebx, 2

mov ecx, %1

mov edx, %2

int 80h

%endmacro

%macro Write 2

mov eax, 4

mov ebx, 1

mov ecx, %1

mov edx, %2

int 80h

%endmacro

%macro fibo 1

mov eax, 0

mov ebx, 1

mov ecx, %1

back:

mov edx, eax

add edx, ebx

;add eax, '0'

mov [temp], eax

pusha

call display

Write disp\_buffer, 2

Write spc, spclen

popa

mov eax, ebx

mov ebx, edx

loop back

%endmacro

section .data

msg1 db 'Enter a Number : ' ;message to enter a number

len1 equ $ -msg1 ;message length

msg2 db 0xa,'Fibonacii series : ' ;message to display fibonacci series

len2 equ $ -msg2 ;message 2 length

spc db ' ' ;space

spclen equ $ -spc

newLine db 13,10

newLen equ $-newLine

section .bss

num resb 2 ;num variable

temp resb 2 ;temp variable

disp\_buffer resb 2 ;display buffer

section .text

global \_start

\_start:

Write msg1, len1 ;write the message on to the screen

read num, 2 ;read the num and store in num

call convert ;call the convert procedure

mov [num], ebx ;mov ebx into num

Write msg2, len2 ;write the message to display fibonacci series

fibo [num] ;fibo macro

Write newLine,newLen

mov eax,1 ;system call

int 80h ;call kernel

convert:

mov esi, num ;store num in esi

mov ecx, 02h ;move 02 in to the counter

xor eax, eax ;clear eax

xor ebx, ebx ;clear ebx

loop\_start:

rol bl, 04h ;rol bl by 4

mov al, [esi] ;mov the current esi content in to al

cmp al, 39h ;cmp al with 39 or digit 9

jbe sub ;jump to sub if below or equal to

sub al, 07h ;sub al with 07 otherwise

sub:

sub al, 30h ;sub al with 30 to convert to decimal

add bl, al

inc esi ;increment esi pointer

loop loop\_start ;loop to loop\_start

ret

display:

mov bl, [temp] ;move contents of temp into bl

mov cl, 2 ;move 2 to cl to be used as a counter

mov edi, disp\_buffer ;mov the display buffer address into edi

next\_digit:

rol bl, 4 ;rotate left by 4 bits

mov al, bl ;mov contents of bl holding temp into al

and al, 0Fh ;and al with 0Fh to clear the left portion

cmp al, 09h ;and compare with 09h

jbe add ;if below or equal than jump to add

add al, 07h ;else add al with 07h

add:

add al, 30h ;add al with 30h

mov [edi], al ;

inc edi

loop next\_digit

ret

OUTPUT:

