**Assignment number: 9**

**Subject: MICROPROCESSOR LAB**

Name: ***RIA MITTAL***

Class: ***SECOND YEAR ENGINEERING***

Division: ***B***

Roll no: ***222008***

Batch: ***B1***

**PROBLEM STATEMENT:**

Write x86 ALP to find the factorial of a given integer number on a command line by using recursion. Explicit stack manipulation is expected in the code.

**Code:**

%macro scall 4 ;common macro for input/output

mov rax,%1

mov rdi,%2

mov rsi,%3

mov rdx,%4

syscall

%endmacro

section .data

num db 00h

msg db "Factorial is : "

msglen equ $-msg

msg1 db "\*\*\*\*\*Program to find Factorial of a number\*\*\*\*\* ",0Ah

db "Enter the number : ",

msg1len equ $-msg1

msgth db 0Ah,"Thank you for using program",0Ah

msgthlen equ $-msgth

zerofact db " 00000001 "

zerofactlen equ $-zerofact

section .bss

dispnum resb 16

result resb 4

temp resb 3

section .text

global \_start

\_start:

scall 1,1,msg1,msg1len

scall 0,0,temp,3 ;accept number from user

call convert ;convert number from ascii to hex

mov [num],dl

scall 1,1,msg,msglen

xor rdx,rdx

xor rax,rax

mov al,[num] ;store number in accumulator

cmp al,01h

jbe endfact

xor rbx,rbx

mov bl,01h

call factr ;call factorial procedure

call display

call exit

endfact:

scall 1,1,zerofact,zerofactlen

call exit

factr: ;recursive procedure

cmp rax,01h

je retcon1

push rax

dec rax

call factr

retcon:

pop rbx

mul ebx

jmp endpr

retcon1: ;if rax=1 return

pop rbx

jmp retcon

endpr:

ret

display: ; procedure to convert hex to ascii

mov rsi,dispnum+15

xor rcx,rcx

mov cl,16

cont:

xor rdx,rdx

xor rbx,rbx

mov bl,10h

div ebx

cmp dl,09h

jbe skip

add dl,07h

skip:

add dl,30h

mov [rsi],dl

dec rsi

loop cont

scall 1,1,dispnum,16

ret

convert: ;procedure to convert ascii to hex

mov rsi,temp

mov cl,02h

xor rax,rax

xor rdx,rdx

contc:

rol dl,04h

mov al,[rsi]

cmp al,39h

jbe skipc

sub al,07h

skipc:

sub al,30h

add dl,al

inc rsi

dec cl

jnz contc

ret

exit: ;exit system call

scall 1,1,msgth,msgthlen

mov rax,60

mov rdi,0

syscall

ret

