NAME: Priyanka Suresh Salunke

CLASS: SE COMP 1

PRN: F19111151

SOURCE CODE:

%macro print 2 ;to print message on the screen

mov rax,1

mov rdi,1

mov rsi,%1

mov rdx,%2

syscall

%endmacro

%macro gtch 1 ;macro for accept from keyboard

mov rax,0 ;standard input

mov rdi,0 ;system for read

mov rsi,%1 ;input the message

mov rdx,1 ;message length

syscall ;interrupt for 64-bit

%endmacro ;close macro

%macro exitprog 0 ;macro for exit

mov rax,60 ;system for exit

mov rdx,0

syscall ;interrupt for 64-bit

%endmacro ;close macro

section .data

msg db "Program for Arithmetic operations.",10

msglen equ $-msg

msg1 db "Addition is:-.",10

msglen1 equ $-msg1

msg2 db "Subtraction is:-",10

msglen2 equ $-msg2

msg3 db "Multiplication is:-",10

msglen3 equ $-msg3

msg4 db "Division is:-",10

msglen4 equ $-msg4

m1 db 10,"1. ADD",10,"2. SUB",10,"3. MUL",10,"4.DIV",10,"5.Exit",10,10, "Enter your choice (1/2/3/4/5<ENTER>): "

l1 equ $-m1

no1 db 04

no2 db 02

newline db 0xa

section .bss

dispbuff resb 2 ;to store ASCII value

input resb 1

choice resb 1

section .txt

global \_start

\_start:

print msg,msglen

back:

print m1,l1 ;Displaying the first message

gtch input ;To read and discard ENTER key pressed.

mov al, byte[input] ;Get choice

mov byte[choice],al

gtch input ;To read and discard ENTER key pressed.

mov al, byte[choice]

cmp al, '1' ;compare contents of al with 1

je add ;if equal the jump to succ\_add procedure

cmp al, '2' ;compare the contents of al with 2

je sub ;if equal the jump to shft\_add procedure

cmp al, '3' ;compare the contents of al with 2

je multi

cmp al, '4'

je div

cmp al, '5' ;compare the contents of al with 3

jnz back ;if not zero then jump to back

exitprog

add:

mov al,[no1]

mov bl,[no2]

add bl,al

print msg1,msglen1

call disp\_result

ret

sub:

mov bl,[no1]

mov al,[no2]

sub bl,al

print msg2,msglen2

call disp\_result

ret

multi:

print msg3,msglen3

mov bl,[no1]

mov al,[no2]

mul bl

mov bl,al

call disp\_result

ret

div:

print msg4,msglen4

mov al,[no1]

mov bl,[no2]

div bl ;divided by bl

mov bl,al

call disp\_result

ret

disp\_result:

mov rdi,dispbuff

mov rcx,02

dispup1:

rol bl,4

mov dl,bl

and dl,0fh

add dl,30h

cmp dl,39h

jbe dispskip1

add dl,07h

dispskip1:

mov [rdi],dl

inc rdi

loop dispup1

print dispbuff,2

ret

OUTPUT:

