Computer Organisation & Architecture Assignment No. 1

Name: Akash Jivendra Bachhav

Batch: C3

Roll No: 199

Problem Statement:

Write an ALP for 64 bit Arithmetic operations and display the result. Accept the numbers from the user.

Source Code:

```
%macro WRITE 02
mov rax,01
mov rdi,01
mov rsi,%1
mov rdx,%2
syscall
%endmacro

%macro READ 02
mov rax,00
mov rdi,00
mov rsi,%1
mov rdx,%2
syscall
%endmacro

section .data
menu db "1. Addition Of Numbers",10
db "2. Subtraction of Numbers",10
db "3. Multiplication of Numbers",10
db "4. Division of Numbers",10
db "5. Exit",10
db "Enter your choice: ",10
menulen equ $-menu

msg1 db "Enter two numbers: ",10
```

```
len1 equ $-msg1
msg2 db "The addition of Numbers is: ",10
len2 equ $-msg2
msg3 db "The subtraction of Numbers is: ",10
len3 equ $-msg3
msg4 db "The multiplication of Numbersis: ",10
len4 equ $-msg4
msg5 db "The Quotient of Division is: ",10
len5 equ $-msg5
msg6 db "The Remainder of Division is: ",10
len6 equ $-msg6
msg7 db "Wrong choice: ",10
len7 equ $-msg7
msg8 db "",10
len8 equ $-msg8

section .bss
a resq 1
b resq 1
c resq 1
d resq 1
char_buff resb 17
actl resq 1
choice resb 02

section .text
global _start
_start:
WRITE msg1,len1
READ char_buff,17
call accept
mov[a],rbx
READ char_buff,17
call accept
mov[b],rbx

printmenu:
WRITE msg8,len8
WRITE menu,menulen
READ choice,02
cmp byte[choice],31H
je addition
cmp byte[choice],32H
je subtraction
cmp byte[choice],33H
```

```
je multiplication
cmp byte[choice],34H
je division
cmp byte[choice],35H
je exitcode

WRITE msg7,len7
jmp printmenu

addition:
mov rax,[a]
add rax,[b]
mov [c],rax
WRITE msg2,len2
mov rbx,[c]
call display
jmp printmenu

subtraction:
mov rax,[a]
sub rax,[b]
mov [c],rax
WRITE msg3,len3
mov rbx,[c]
call display
jmp printmenu

multiplication:
mov rax,qword[a]
mul qword[b]
mov [c],rdx
mov [d],rax
WRITE msg4,len4
mov rbx,[c]
call display
mov rbx,[d]
call display
jmp printmenu

division:
mov rdx,00
mov rax,qword[a]
div qword[b]
mov [c],rax
mov [d],rdx
```

```
WRITE msg5,len5
mov rbx,[c]
call display
WRITE msg6,len6
mov rbx,[d]
call display
jmp printmenu
exitcode:
mov rax,60
mov rsi,00
syscall

accept: dec rax
mov [actl],rax
mov rbx,00
mov rsi,char_buff

up:shl rbx,04H
mov rdx,00H
mov dl,byte[rsi]
cmp dl,39H
jbe sub30
sub dl,07H

sub30:sub dl,30H
add rbx,rdx
inc rsi
dec qword[actl]
jnz up
ret
display:mov rcx,16
mov rsi,char_buff

above:rol rbx,04H
mov dl,bl
and dl,0FH
cmp dl,09H
jbe add30
add dl,07H

add30:add dl,30H
mov byte[rsi],dl
inc rsi
dec rcx
jnz above
```

WRITE char_buff,16
Ret

Output Screen: