

# Code:-

---

```
%macro display 2
mov rax,1
mov rdi,1
mov rsi,%1
mov rdx,%2
syscall
%endm
```

```
%macro input 2
mov rax,0
mov rdi,0
mov rsi,%1
mov rdx,%2
syscall
%endm
```

```
global _start
```

```
section .data
```

```
msg db 0dh,0ah,"result of addition is:"          ;Taking input from user
msg_len equ $-msg
```

```
msg1 db 0dh,0ah,"result of subtraction is:"
msg1_len equ $-msg1
```

```
msg2 db 0dh,0ah,"result of multiplication is:"
msg2_len equ $-msg2
```

```
msg3 db 0dh,0ah,"result of division is:Quotionent : "
msg3_len equ $-msg3
```

```
msg4 db 0dh,0ah,"result of division is:Remainder : "
msg4_len equ $-msg4
```

```
msg5 db 0dh,0ah,"Enter your choice for 1:addition 2:subtraction 3:multiplication
4:division : "
msg5_len equ $-msg5
```

```
msg6 db 0dh,0ah,"Error :"  
msg6_len equ $-msg6
```

```
num1 dq 012345678h  
num2 dq 010h
```

```
section .bss
```

```
add_result resq 1  
add_result_ascii resq 2  
add_carry resb 1  
add_carry_ascii resb 1  
sub_result resq 1  
mulrdx_result resq 1  
mulrax_result resq 1  
quo_result resq 1  
rem_result resq 1  
choice resb 2
```

```
section .text
```

```
_start:
```

```
display msg5,msg5_len  
input choice,2
```

```
mov rsi,num1  
mov rax,[rsi]  
mov rbx,[num2]
```

```
mov cl,[choice]  
cmp cl,31h  
jne dn_ch
```

```
call addition  
jmp ext
```

```
dn_ch:  
cmp cl,32h
```

jne dn\_ch1

call subtraction  
jmp ext

dn\_ch1:  
cmp cl,33h  
jne dn\_ch2

call multiplication  
jmp ext

dn\_ch2:  
cmp cl,34h  
jne dn\_error

mov rdx,0  
call division  
jmp ext

dn\_error:  
display msg6,msg6\_len

ext: mov rax,60  
mov rdi,0  
syscall

division:  
div rbx  
mov [rem\_result],rdx  
mov [quo\_result],rax

display msg3,msg3\_len  
mov rax,[quo\_result]  
call disp\_result  
display add\_result\_ascii,10h

display msg4,msg4\_len  
mov rax,[rem\_result]  
call disp\_result

**display add\_result\_ascii,10h**

**ret**

**addition:**

**mov cl,00**

**add rax,rbx**

**jnc dn1**

**inc cl**

**dn1: mov [add\_result],rax**

**mov [add\_carry],cl**

**display msg,msg\_len**

**mov cl,[add\_carry]**

**add cl,30h**

**mov [add\_carry\_ascii],cl**

**display add\_carry\_ascii,1**

**mov rax,[add\_result]**

**call disp\_result**

**display add\_result\_ascii,10h**

**ret**

**subtraction:**

**sub rax,rbx**

**mov [sub\_result],rax**

**display msg1,msg1\_len**

**mov rax,[sub\_result]**

**call disp\_result**

**display add\_result\_ascii,10h**

**ret**

```

multiplication:
mul rbx
mov [mulrax_result],rax
mov [mulrdx_result],rdx

display msg2,msg2_len

mov rax,[mulrdx_result]
call disp_result
display add_result_ascii,10h
mov rax,[mulrax_result]
call disp_result
display add_result_ascii,10h

ret

```

```

disp_result:
mov cl,10h
mov rdi,add_result_ascii

;RAX= 1234567812345678
up1: rol rax,4
;RAX= 2345678123456781
mov rbx,rax
and al,0fh
;RAX= 0000000000000001
cmp al,09
ja dn2
add al,30h
jmp dn3
dn2: add al,37h
dn3: mov [rdi],al
inc rdi
mov rax,rbx
dec cl
jnz up1
ret

```

---

**Output:-**

</> Code

≡ Input

>\_ Output

1

1

</> Code

≡ Input

>\_ Output

▶ Run

📄 Save

```
Enter your choice for 1:addition 2:subtraction 3:multiplication 4:division :  
result of addition is:00000000012345688  
[Program exited with exit code 0]
```

---

**Nikhil Vinod Khodake**  
**9022**  
**SE Computer**  
**MP Assignments**