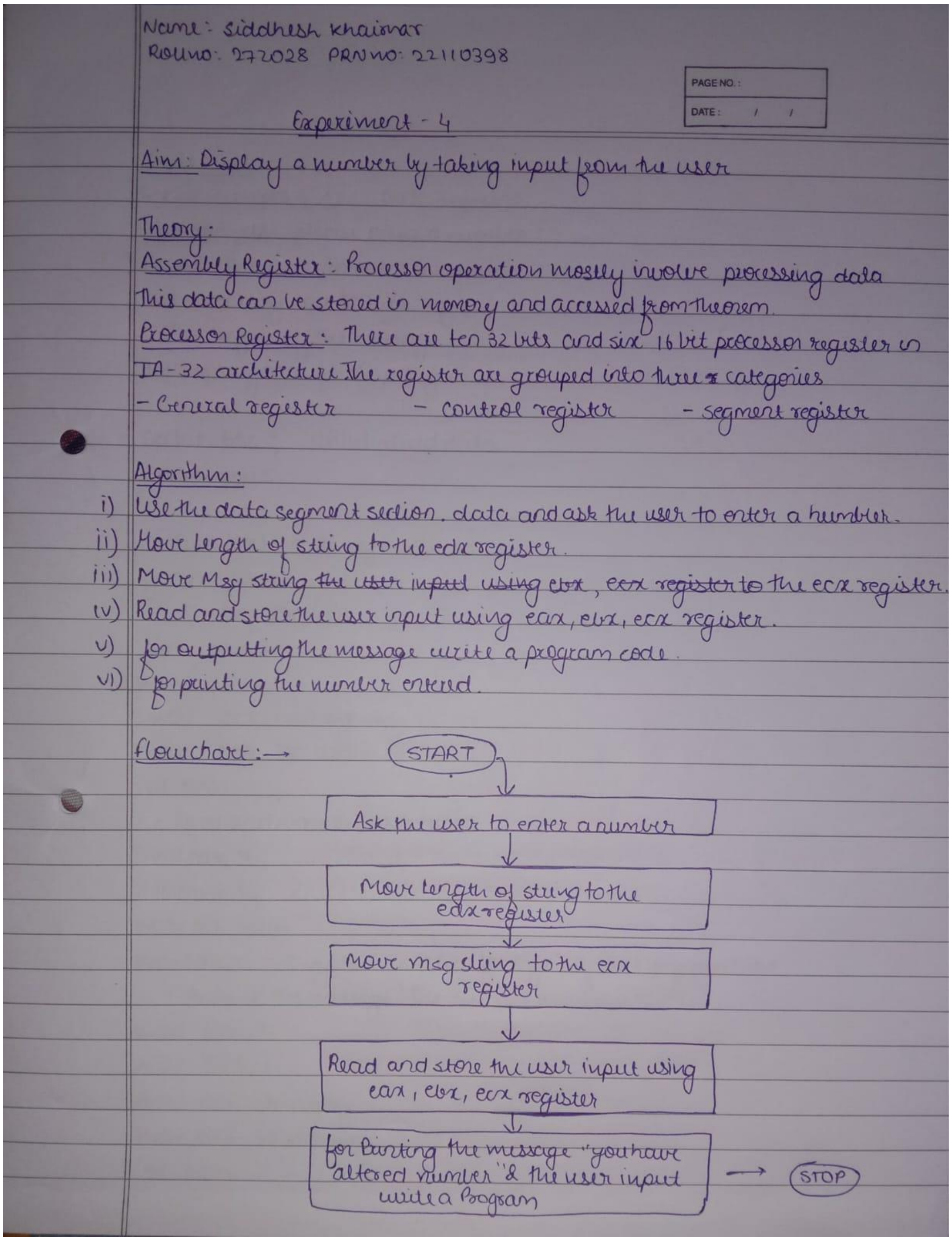
MP Practical- 4

Name- Siddhesh Dilip Khairnar

Roll No.- 272028

Batch- B2



Name: siddhesh khairnar
Rollno: 272028 PRN no: 22110398

PAGE NO.:
DATE: / /

Experiment - 4

Aim: Display a number by taking input from the user

Theory:
Assembly Register: Processor operation mostly involve processing data This data can be stored in memory and accessed from theorem.
Processor Register: There are ten 32 bits and six 16 bit processor register in IA-32 architecture The register are grouped into three ~~x~~ categories
- General register - control register - segment register

Algorithm:
i) Use the data segment section. data and ask the user to enter a number.
ii) Move length of string to the edx register.
iii) Move Msg string ~~the user input using ebx~~, ecx register to the ecx register.
iv) Read and store the user input using eax, ebx, ecx register.
v) for outputting the message write a program code.
vi) for printing the number entered.

Flowchart:→

START
↓
Ask the user to enter a number
↓
Move length of string to the edx register
↓
Move msg string to the ecx register
↓
Read and store the user input using eax, ebx, ecx register
↓
for Printing the message "you have altered number" & the user input write a Program → STOP

PAGE NO.:
DATE: / /

code :→

```
section data  ;    Data segment
user msg db: 'please enter a number : ';
  Ask the user to enter a number
len user msg equ $- user mg;
  The length of the message
lendispmsgequ $-dispmsg

sector. bss ;  unintialized data
 num resb 5

section.text; code segment
global_start.
   ; user prompt
mov eax, 4
mov ebx, 1
mov ecx, user~~msg~~ msg
mov edx, len user msg
int 80h
  ; Read and store the user input
mov eax, 3
mov ebx, 2
mov ecx, num
mov edx, 5  ; 5 bytes (numeric 1 for sign) of that information
  ; Output the message 'The entered number is:'
mov eax, 4
mov ebx, 1
mov ecx, dispmsg
mov edx, Lendismsg
int 80h
```

PAGE NO.:
DATE: / /

```
 ; Output the number entered
mov eax, 4
mov ebx, 1
mov ecx, num
mov edx, 5
int 80h
; Exist code
mov edx, 1
mov ebx, 0
int 80h
```

Conclusion: Soure hearnt how to accept a number and print it in assembly language using various register.

Code:

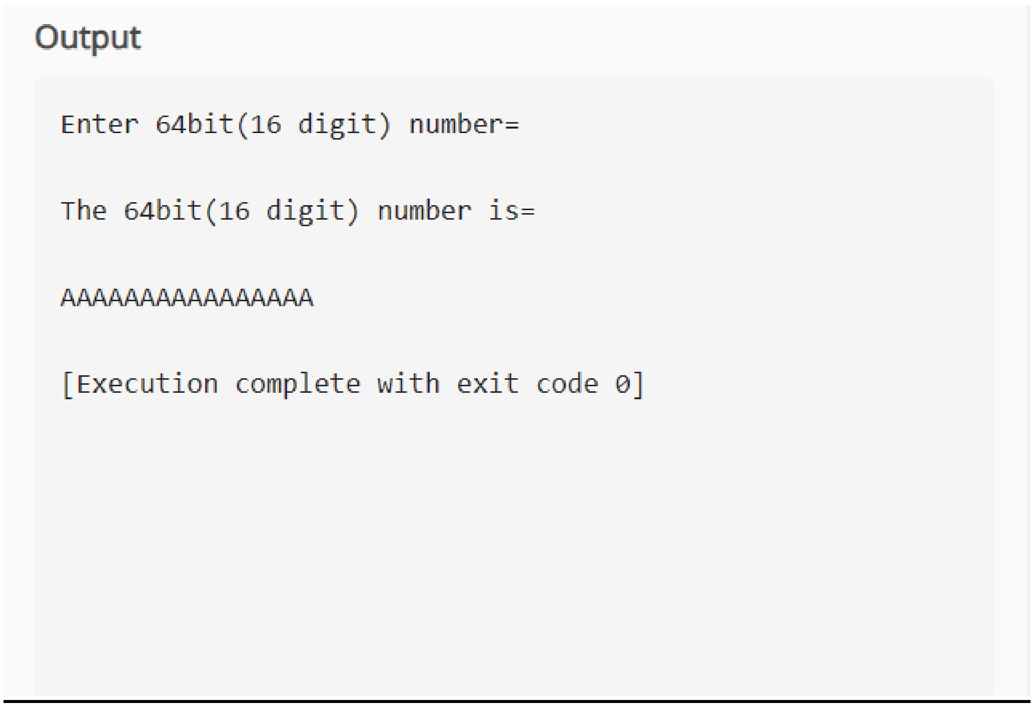
```
%macro scall 4
        mov rax,%1
        mov rdi,%2
        mov rsi,%3
        mov rdx,%4
        syscall
%endmacro

section .data

        m1 db "Enter 64bit(16 digit) number=",10d,13d
        l1 equ $-m1

        m2 db "The 64bit(16 digit) number is=",10d,13d
        l2 equ $-m2

        m3 db " ",10
        l3 equ $-m3


section .bss
        num resb 20
        array resb 200
        char_ans resb 16

section .text
global _start
```

```
_start:

        scall 1,1,m1,l1

        scall 0,0,num,17

        call accept_proc



        mov rbp,array

        mov qword[rbp],rbx



;/********Display 64BIT Number*********/

        scall 1,1,m2,l2

        mov rbx,array

        mov rax,[rbx]

        call display_proc
```

```
;/*********EXIT**********/
        mov rax,60
        mov rdi,0
        syscall


;********ACCEPT PROCEDURE *********
accept_proc:
        mov rsi,num
        mov rbx,0
        mov rax,0
        mov rcx,16
back:
        rol rbx,04
        mov al,[rsi]

        cmp al,39h
        jbe next
        sub al,07h
next:

        sub al,30h
        add rbx,rax

        inc rsi
        dec rcx
        jnz back
ret
```

```
;/***********Display Procedure***********/

display_proc:

        mov rbp,char_ans
        mov rcx,16

up3:
        rol rax,04
        mov dl,al

        and dl,0Fh

        cmp dl,09h
        jbe next1

        add dl,07h
next1:
        add dl,30h

        mov [rbp],dl

        inc rbp
        dec rcx

        jnz up3

        scall 1,1,char_ans,17
        scall 1,1,m3,l3

ret
```

Output:



Output

```
Enter 64bit(16 digit) number=

The 64bit(16 digit) number is=

AAAAAAAAAAAAAAAA

[Execution complete with exit code 0]
```