

ASM LAB ASSIGNMENT 1

Name: Hrithvik Kondalkar

Roll: 002211001088

1. Write an Assembly Language Program to add two byte integers and store the result in DX register.

```
.model small
.stack 100h
.data
    var1 db 02h
    var2 db 06h
.code
main proc
    mov ax, @data
    mov ds, ax

    mov dl, var1

    add dl, var2
    add dl, 48

    mov ah, 02h
    int 21h

    mov ah, 4ch
    int 21h
main endp
end main
```

2. Write an 8086 Assembly Language Program to subtract two 8-bit signed integers. The numbers can be stored in the data segment.

```
.model small
.stack 100h
.data
    var1 db 9
    var2 db 3
.code
main proc
    mov ax, @data
    mov ds, ax

    mov dl, var1

    sub dl, var2
    add dl, 48

    mov ah, 02h
    int 21h

    mov ah, 4ch
    int 21h
main endp
end main
```

3. Write an Assembly Language Program to print your name, which is stored in memory as a string.

```
.model small
.stack 100h
.data
    msg1 db 'hrithvik$'
.code
main proc
    mov ax, @data
    mov ds, ax

    mov dx, offset msg1

    mov ah, 09h
    int 21h

    mov ah, 4ch
    int 21h
main endp
end main
```

4. Write an Assembly Language Program to reverse a string using stack and display the result.

```
.model small
.stack 100h
.data
    string db "hrithvik$"
.code
main proc
    mov ax, @data
    mov ds, ax
    mov si, offset string
    mov cx, 8

    tostack:
        push [si]
        inc si
    loop tostack

    mov cx, 8
    mov ah, 02h

    print:
        pop dx
        int 21h
    loop print

    mov ah, 4ch
    int 21h

main endp
end main
```

5. Write an 8086 Assembly Language Program which will ask for a number and the no. will be taken from keyboard. Print the number in decimal, binary and hexadecimal format.

<pre> strprint macro str mov dx, offset str mov ah, 09h int 21h endm .model small .stack 100h .data innumstr db 100 dup("\$") num db 0 msg db "number in decimal : \$" msgbin db "binary : \$" msghex db "hexadeximal : \$" .code main proc mov ax, @data mov ds, ax strprint msg mov si, offset innumstr mov cl, 10 mov dl, 0 getnum: mov ah, 01h int 21h cmp al, 13 je binary mov [si], al sub al, 48 mov dl, al xor ax, ax mov al, num mul cl mov num, al add num, dl inc si jne getnum </pre>	<pre> binary: xor ax, ax mov cl, 2 mov al, num mov bl, 0 tobinary: div cl mov bh, al xor al, al add ah, 48 mov al, ah xor ah, ah push ax inc bl mov al, bh cmp al, 0 jne tobinary printbin: call nextline strprint msgbin xor cx, cx mov cl, bl mov ah, 02 crtstr: xor dx, dx pop dx int 21h loop crtstr </pre>	<pre> hex: xor ax, ax mov cl, 16 mov al, num mov bl, 0 tohex: div cl xor dx, dx mov dl, ah cmp ah, 10 jge hexi add dl, 48 jmp conthex hexi: add dl, 87 conthex: push dx inc bl xor ah, ah cmp al, 0 jne tohex printhex: call nextline strprint msghex xor cx, cx xor dx, dx mov cl, bl mov ah, 02h poploop: pop dx int 21h loop poploop mov ah, 4ch int 21h main endp nextline proc mov dx, 10 mov ah, 2 int 21h mov dx, 13 int 21h ret nextline endp end main </pre>
--	--	--

6. Now modify the program in Q2 so that it will ask for your name and takes the input from keyboard.

```
.model small
.stack 100h
.data
    str1 db 100 dup('$')
.code
main proc
    mov ax, @data
    mov ds, ax

    mov si, offset str1

    uinp:
        mov ah, 01h
        int 21h
        mov [si], al
        inc si
        cmp al, 13
        jne uinp

    mov dx, offset str1
    mov ah, 09h
    int 21h

    mov ah, 4ch
    int 21h
main endp
end main
```

7. Write an Assembly Language Program to check the length of a given string.

```
.model small
.stack 100h
.data
    string db "hrithvik$"
.code
main proc
    mov ax, @data
    mov ds, ax

    mov si, offset string

    mov dl, '0'
    estring:
        mov al, [si]
        cmp al, '$'
        je finish
        add dl, 1
        inc si
        jmp estring

    finish:
        mov ah, 02h
        int 21h
        mov ah, 4ch
        int 21h
main endp
end main
```

