

**LAB Report**

**COURSE TITLE –** Microprocessor Lab

**COURSE CODE –** CSE 360

***Submitted To***

***Md. Ismail***

*Lecturer of UITS*

***Submitted By***

*Md. Shoyaif Rahman (****0432310005101050****)*

***Semester:*** *Autumn-2025 (6th)*

***Department:*** *CSE*

***Batch:*** *53*

***Section:****6A2*

Date: 03/09/25

Assembly Language Lab Report

# Lab report: 04

# Experiment Name: Write a program to print right angle triangle and Equilateral triangle

# Process:

**Printing right angle triangle:**

1. Start the program and set up stack and data segment.

2. Read a single-digit number from the user and convert it from ASCII to numeric.

3. Move the cursor to a new line for output clarity.

4. Initialize loop registers (BL = 1, CX for counting).

5. Outer loop: repeat for each row until the user-defined height.

6. Inner loop: print '\*' characters for the current row.

7. Increment BL to increase stars for the next row.

8. Move to a new line after each row.

9. Repeat until all rows are printed.

10. Exit the program.

**Printing Equilateral triangle:**

1. Start the program and initialize the data segment.

2. Read a single-digit number from the user and convert it from ASCII to numeric.

3. Move the cursor to a new line for output clarity.

4. Initialize loop registers (BL = 1, CL = VAR for pyramid height).

5. Outer loop: repeat for each row of the pyramid.

6. Print spaces before stars to center the pyramid.

7. Print stars for the current row (odd number of stars: 1, 3, 5, ...).

8. Move to a new line after each row.

9. Increment BL to increase the number of stars for the next row.

10. Repeat until all rows are printed.

11. Exit the program.

## 2. Implementation (Program Code – ASM)

**1.Printing right angle triangle:**

.model small

.stack 100h

.data

VAR DB ?

.code

MAIN PROC

MOV AH, 1

INT 21H

SUB AL, 48

MOV VAR, AL

MOV AH,2

MOV DL,10

INT 21H

MOV DL,13

INT 21H

MOV CX,0

MOV CL, VAR

MOV BL, 1

TOP:

MOV CX,BX

LEVEL1:

CMP BL,VAR

JG EXIT

MOV AH, 2

MOV DL, '\*'

INT 21H

LOOP LEVEL1

INC BL

MOV AH,2

MOV DL,10

INT 21H

MOV DL,13

INT 21H

LOOP TOP

EXIT:

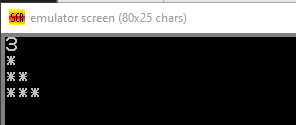
MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

**output:**

****

**2.Printing Equilateral triangle:**

.model small

.stack 100h

.data

VAR DB ?

.code

MAIN PROC

mov ax,@data

mov ds,ax

mov ah,1

int 21h

sub al,48

mov var,al

mov ah,2

mov dl,0Dh

int 21h

mov dl,0Ah

int 21h

mov cl,var

mov ch,0

mov bl,1

outer\_loop:

mov al,var

sub al,bl

mov bh,al

print\_space:

cmp bh,0

je print\_star

mov ah,2

mov dl,' '

int 21h

dec bh

jmp print\_space

print\_star:

mov al,bl

shl al,1

dec al

mov bh,al

star\_loop:

cmp bh,0

je row\_done

mov ah,2

mov dl,'\*'

int 21h

dec bh

jmp star\_loop

row\_done:

mov ah,2

mov dl,0Dh

int 21h

mov dl,0Ah

int 21h

inc bl

loop outer\_loop

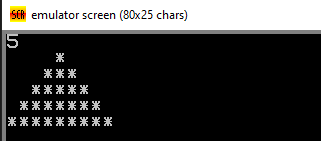
mov ah,4Ch

int 21h

MAIN ENDP

end MAIN

**output:**



**3.Result**

## The first program prints a right-angled triangle of stars with the height equal to the number entered by the user, where the number of stars increases by one in each row. The second program prints a centered pyramid of stars, also based on the user input, with each row containing an odd number of stars (1, 3, 5, …) to form the pyramid shape. Both programs effectively demonstrate the use of loops, ASCII conversion, and DOS interrupt 21h to display characters on the screen.

## 4. Conclusion

Both programs successfully demonstrate the use of loops, nested loops, and ASCII manipulation in 8086 assembly language to print patterns on the screen. The first program illustrates creating a right-angled triangle by incrementally increasing the number of stars per row, while the second program shows how to form a centered pyramid with each row containing an odd number of stars. These programs reinforce concepts such as loop control, arithmetic operations, and using DOS interrupt 21h for character output, highlighting fundamental techniques for pattern generation in assembly language.