Assembly Programs:

1. Write a AVL too print “Hello World” on screen

section .data

hello db 'Hello World', 10

hellolen equ $-hello

section .text

global \_start

\_start:

mov eax, 4

mov ebx, 1

mov ecx, hello

mov edx, hellolen

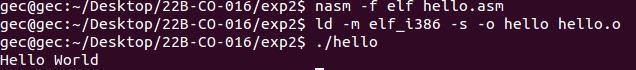
int 80h

mov eax, 1

mov ebx, 0

int 80h

**Output:**



2. Display “Hello World” along with your name

section .data

hello db 'Hello world', 10

hellolen equ $-hello

name db 'Divyam', 10

namelen equ $-name

section .text

global \_start

\_start:

mov eax, 4

mov ebx, 1

mov ecx, hello

mov edx, hellolen

int 80h

mov eax, 4

mov ebx, 1

mov ecx, name

mov edx, namelen

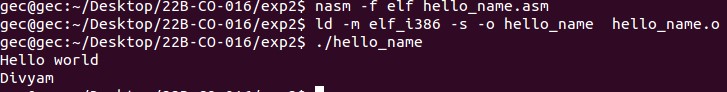
int 80h

mov eax, 1

mov ebx, 0

int 80h

**Output:**



3. Write an AVL to display 9 stars using times directive

section .data

star times 9 db '\*'

newline db 10,0

section .text

global \_start

\_start:

mov eax, 4

mov ebx, 1

mov ecx, star

mov edx, 10

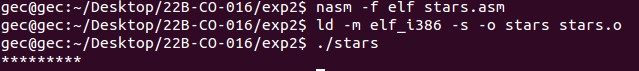
int 80h

mov eax, 1

mov ebx, 0

int 80h

**Output:**



4. write an AVL to display two strings on screen using the equ directive

close equ 0

write equ 4

exit equ 1

section .data

string1 db 'Hello', 10

s1len equ $-string1

string2 db 'World', 10

s2len equ $-string2

section .text

global \_start

\_start:

mov eax, write

mov ebx, close

mov ecx, string1

mov edx, s1len

int 80h

mov eax, write

mov ebx, close

mov ecx, string2

mov edx, s2len

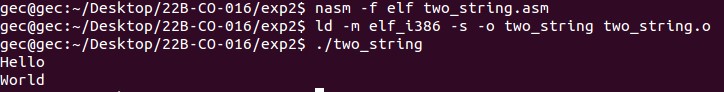
int 80h

mov eax, exit

mov ebx, close

int 80h

**Output:**



5. Write an AVL to replce a word in a given string

section .data

name db "Dr. Divyam",10

namelen equ $-name

section .text

global \_start

\_start:

mov eax, 4

mov ebx, 1

mov ecx, name

mov edx, namelen

int 80h

mov[name], dword "Mr. "

mov eax, 4

mov ebx, 1

mov ecx, name

mov edx, namelen

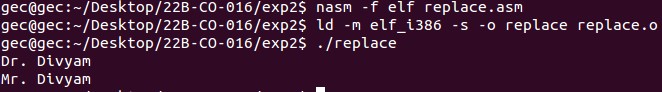
int 80h

mov eax, 1

mov ebx, 0

int 80h

**Output:**



6. Write an AVL to read a number from the keyboard and display it on the screen.

section .data

prompt db "Enter a number : "

promptlen equ $-prompt

section .bss

num: resb 10

section .text

global \_start

\_start:

mov eax, 4

mov ebx, 1

mov ecx, prompt

mov edx, promptlen

int 80h

mov eax, 3

mov ebx, 1

mov ecx, num

mov edx, 10

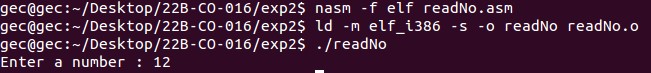
int 80h

mov eax, 1

mov ebx, 0

int 80h

**Output:**



7. Write an AVL to read a name as a string by taking input

section .data

prompt db "Enter a name : "

promptlen equ $-prompt

section .bss

name: resb 10

section .text

global \_start

\_start:

mov eax, 4

mov ebx, 1

mov ecx, prompt

mov edx, promptlen

int 80h

mov eax, 3

mov ebx, 1

mov ecx, name

mov edx, 20

int 80h

mov eax, 4

mov ebx, 1

mov ecx, name

mov edx, 20

int 80h

mov eax, 1

mov ebx, 0

int 80h

**Output:**

****

8. Display two different numbers on the screen using the equ directive

section .data

prompt db "Enter two numbers : ",10

promptlen equ $-prompt

result db "Numbers entered : ",10

resultlen equ $-result

section .bss

num1: resb 5

num2: resb 5

section .text

global \_start

\_start:

mov eax, 4

mov ebx, 1

mov ecx, prompt

mov edx, promptlen

int 80h

mov eax, 3

mov ebx, 1

mov ecx, num1

mov edx, 5

int 80h

mov eax, 3

mov ebx, 1

mov ecx, num2

mov edx, 5

int 80h

mov eax, 4

mov ebx, 1

mov ecx, result

mov edx, resultlen

int 80h

mov eax, 4

mov ebx, 1

mov ecx, num1

mov edx, 5

int 80h

mov eax, 4

mov ebx, 1

mov ecx, num2

mov edx, 5

int 80h

mov eax, 1

mov ebx, 0

int 80h

**Output:**

