2A] Write an assembly language program to display “Hello World” on screen.

Program:

SECTION .data

hello db 'HelloWorld',10

hellolen equ $-hello

SECTION .text

GLOBAL \_start

\_start:

mov eax,4

mov ebx,1

mov ecx,hello

mov edx,hellolen

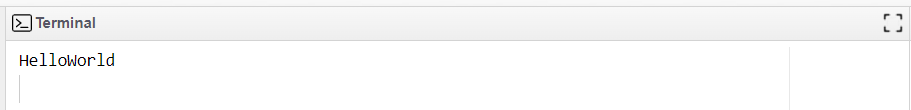
int 80h

mov eax,1

mov ebx,0

int 80h

Output:



2B] Write an assembly language program to display 9 stars using array (times directive).

Program:

SECTION .data

stars times 9 db '\*'

SECTION .text

GLOBAL \_start

\_start:

mov eax,4

mov ebx,1

mov ecx,stars

mov edx,9

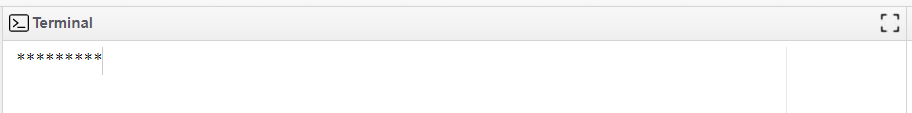
int 80h

mov eax,1

mov ebx,0

int 80h

Output:



2C]

Program:

SECTION .data

hello db 'HelloWorld',10

name db 'Jeevesh',10

hellolen equ $-hello

namelen equ $-name

SECTION .text

GLOBAL \_start

\_start:

mov eax,4

mov ebx,1

mov ecx,hello

mov edx,hellolen

int 80h

mov ecx,name

mov edx,namelen

int 80h

mov eax,1

mov ebx,0

int 80h

Output:



2D]

Program:

SECTION .data

name db 'ABC DEF',10

len equ $-name

text1 db 'Original string:'

len1 equ $-text1

text2 db 'New string:'

len2 equ $-text2

SECTION .text

global \_start

\_start:

mov eax,4

mov ebx,1

mov ecx,text1

mov edx,len1

int 0x80

mov eax,4

mov ebx,1

mov edx,len

mov ecx,name

int 0x80

mov [name], dword 'XYZ '

mov eax,4

mov ebx,1

mov ecx,text2

mov edx,len2

int 0x80

mov ebx,1

mov eax,4

mov ecx,name

mov edx,len

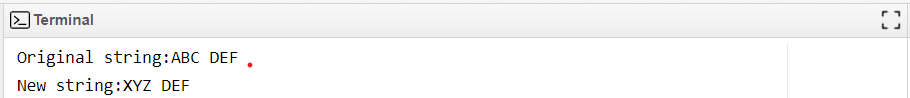
int 0x80

mov eax,1

mov ebx,0

int 0x80

Output:



2E]

Program:

section .text

input db 'Enter a number:'

i\_len equ $-input

output db 'Number entered is:'

o\_len equ $-output

section .bss

num resb 4

section .text

global \_start

\_start:

mov eax,4

mov ebx,1

mov ecx,input

mov edx,i\_len

int 0x80

;Input number

mov eax,3

mov ebx,2

mov ecx,num

mov edx,4

int 0x80

mov eax,4

mov ebx,1

mov ecx,output

mov edx,o\_len

int 0x80

;Output number

mov eax,4

mov ebx,1

mov ecx,num

mov edx,4

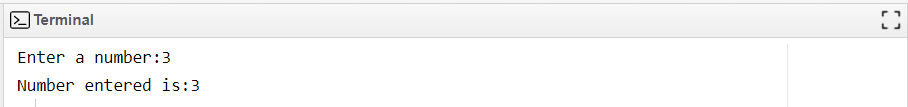
int 0x80

mov eax,1

mov ebx,0

int 0x80

Output:



2F]

Program:

section .text

input db 'Enter the name:'

i\_len equ $-input

output db 'Name entered is:'

o\_len equ $-output

section .bss

len resw 20

section .text

global \_start

\_start:

mov eax,4

mov ebx,1

mov ecx,input

mov edx,i\_len

int 0x80

;Input name

mov eax,3

mov ebx,2

mov ecx,len

int 0x80

mov eax,4

mov ebx,1

mov ecx,output

mov edx,o\_len

int 0x80

;Output name

mov eax,4

mov ebx,1

mov ecx,len

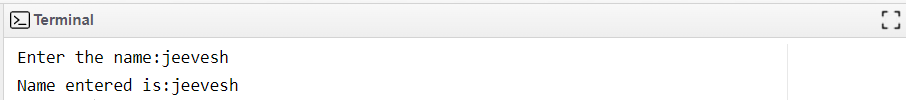
int 0x80

mov eax,1

mov ebx,0

int 0x80

Output:



2G]

Program:

section .data

num1 equ 2

num2 equ 7

msg db "The numbers are: ", 0

buffer times 2 db 0

section .text

global \_start

\_start:

mov eax, 4

mov ebx, 1

mov ecx, msg

mov edx, 17

int 0x80

; convert number1 to string

mov eax, num1

mov ebx, 10

mov edx, 0

div ebx

add edx, '0'

mov byte [buffer], dl

mov byte [buffer+1], ' '

mov esi, buffer

mov ecx, esi

sub ecx, buffer

add ecx, 2

; print number1 string

mov eax, 4

mov ebx, 1

mov edx, ecx

mov ecx, buffer

int 0x80

; convert number2 to string

mov eax, num2

mov ebx, 10

mov edx, 0

div ebx

add edx, '0'

mov byte [buffer], dl

mov esi, buffer

mov ecx, esi

sub ecx, buffer

add ecx, 2

; print number2 string

mov eax, 6

mov ebx, 1

mov edx, ecx

mov ecx, buffer

int 0x80

mov eax, 1

mov ebx, 0

int 0x80

Output:

