*Name: Priyanka Suresh Salunke*

*Class: SE COMP 1*

*Roll no.: 70*

*PRN: F19111151*

***SOURCE CODE:***

%macro print 2 ;macro declaration with 4 parameters

mov rax,1 ;1st parameter has been moved to rax

mov rdi,1 ;2nd parameter has been moved to rdi

mov rsi,%1 ;3rd parameter has been moved to rsi

mov rdx,%2 ;4th parameter has been moved to rdx

syscall ;Call the Kernal

%endmacro ;end of macro

%macro read 2 ;macro declaration with 4 parameters

mov rax,0 ;read function

mov rdi,0 ;reading from keyboard

mov rsi,%1 ;rsi with buffer to store read data

mov rdx,%2 ;length of data wanted to read

syscall ;Call the Kernal

%endmacro ;end of macro

section .data ;.data begins here

m1 db 10,13,"Enter a string: " ;m1 variable initialised with string

l1 equ $-m1 ;l1 stores length of string m1

m2 db 10,13,"Entered String: " ;m2 variable initialised with string

l2 equ $-m2 ;l2 stores length of string m2

m3 db 10,13,"Length: " ;m3 variable initialised with string

l3 equ $-m3 ;l3 stores length of string m3

section .bss ;.bss begins here

buffer resb 50 ;buffer array of size 50

size equ $-buffer ;size variable to have input

count resb 1 ;to store size of buffer

dispnum resb 2 ;to display 16 digit length

section .text ;.text begins here

global \_start ;moving to \_start label

\_start: ;\_start label

print m1,l1 ;macro call to display m1

read buffer,size ;macro call to input buffer

mov byte[count],al ;length of buffer gets stored in count

print m2,l2 ;macro call to display m2

print buffer,[count] ;macro call to display buffer

call display

Exit:

mov rax,60 ;sys\_exit function

mov rbx,0 ;Sucessful Termination

syscall ;Call the Kernel

display:

mov rdi,dispnum ;rsi points to 16th location of dispnum

mov bl,byte[count] ;rax now stores value of count

mov rcx,4 ;rcx gets initiaised with 4

dec rbx ;decrement the value of rbx

dispup1:

rol bl,4

mov dl,bl

and dl,0fh

add dl,30h

cmp dl,39h

jbe dispskip1

add dl,07h

dispskip1:

mov [rdi],dl

inc rdi

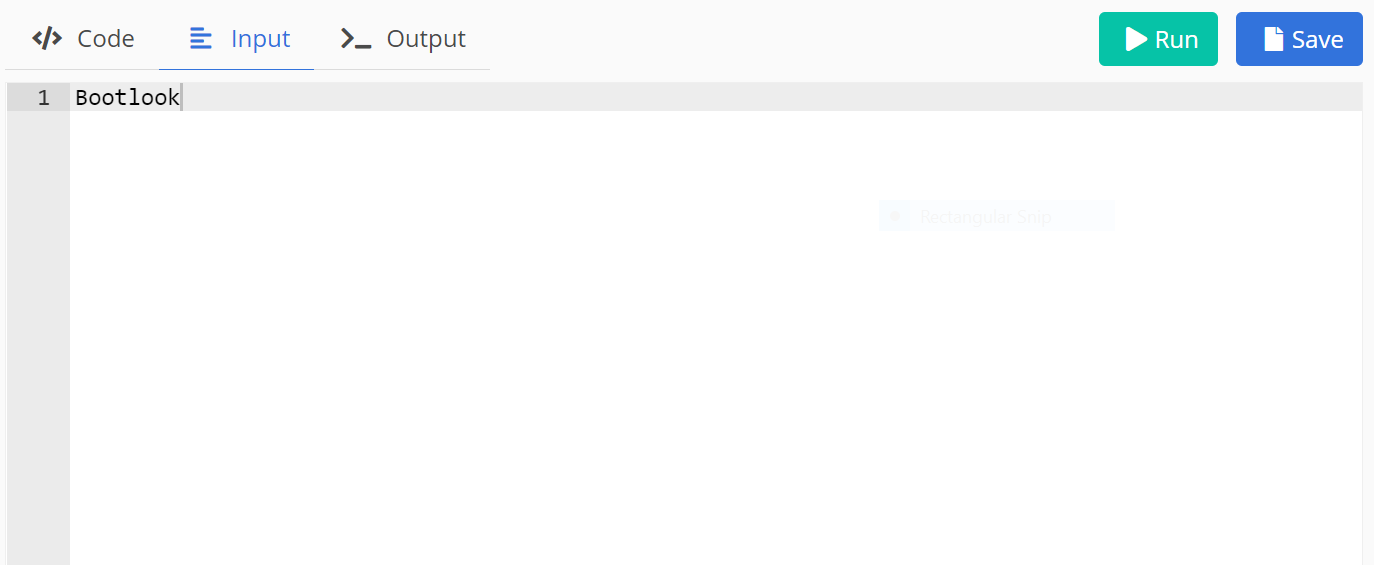
loop dispup1

print m3,l3 ;macro call to display m3

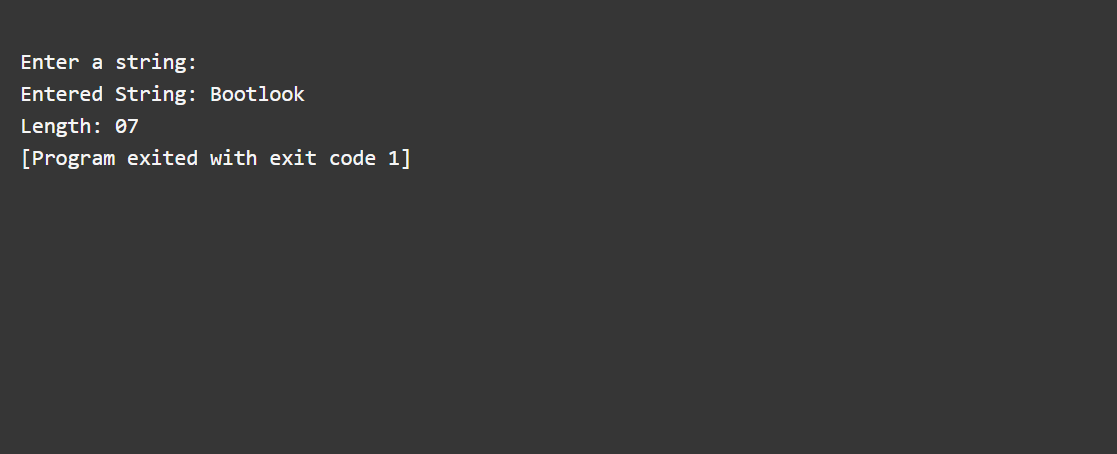
print dispnum,2 ;macro call to display dispnum array

ret

***INPUT:***

******

***OUTPUT:***

******