%macro print 2

mov rax,01

mov rdi,1

mov rsi,%1

mov rdx,%2

syscall

%endmacro

section .data

srcblk db 10h,20h,30h,40h,50h

dstblk db 60h,61h,62h,63h,64h

m0 db 10,13, "Overlapping BDT with string instructions", 10,13

l0 equ $-m0

m1 db 10,13," Source Block: ",10,13

l1 equ $-m1

m2 db 10," Destinition Block Before Transfer: ",10,13

l2 equ $-m2

m3 db 10," Destinition Block After Transfer: ",10,13

l3 equ $-m3

space db " "

newline db 0xa

section .bss

input resb 01

choice resb 01

count resb 01

count1 resb 01

section .text

global \_start

\_start:

menu:

print m0,l0

print m1,l1

mov rsi,srcblk

call disp\_block

print m2,l2

mov rsi,dstblk

call disp\_block

print m3,l3

cld

mov rcx,02

mov rsi,srcblk

mov rdi,dstblk

s1: movsb

loop s1

mov rcx,03

mov rsi,srcblk

s2: movsb

loop s2

mov rsi,dstblk

call disp\_block

print newline,1

stop:

mov rax,60

xor rdi,rdi

syscall

disp\_block:

mov rbp,05

back: mov al,[rsi]

push rsi

mov bl,al

call disp\_8

print space,1

pop rsi

inc rsi

dec rbp

jnz back

ret

disp\_8:

mov dl,bl

and dl,0f0h

rol dl,04

cmp dl,09h

jbe skip

add dl,07h

skip:

add dl,30h

mov byte[count],dl

and bl,0fh

cmp bl,09h

jbe skip1

add bl,07h

skip1:

add bl,30h

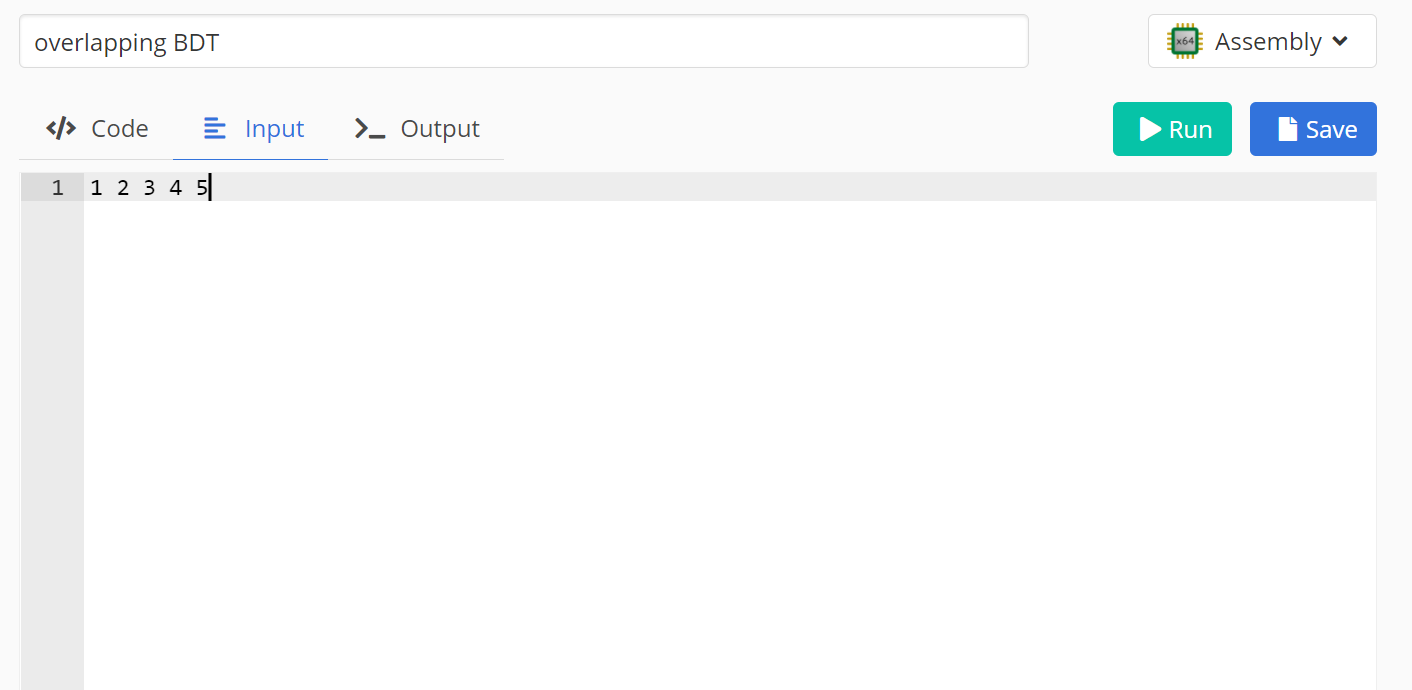
mov byte[count1],bl

print count,01

print count1,01

ret

INPUT



OUTPUT:

