msg1: DC "One\0"

msg2: DC "TwoThree\0"

buffer: DC " "

STACK: EQU 0x100000

lui sp, STACK >> 12

addi sp, sp, 64

addi a1, x0, msg1 # Load address of msg1 into a1

addi a2, x0, msg2 # Load address of msg2 into a2

addi a3, x0, buffer # Load address of buffer into a3

jal x1, appach # Call the concat procedure

addi a1, x0, buffer # Load address of buffer (the concatenated result) into a1

jal x1, lench # Call the lench procedure to find the length of the concatenated string

addi x6, x0, buffer # Load space for output

ecall x0, x6, 4 # Output the concatenated string

ebreak x0, x0, 0 # Finish

appach:

# Append msg1 to buffer

loop1:

lb x5, 0(a1) # Load byte from msg1

beq x5, x0, append\_second # If null terminator, go to append second

sb x5, 0(a3) # Store byte in output buffer

addi a1, a1, 1 # Move to next character in msg1

addi a3, a3, 1 # Move to next position in output buffer

jal x0, loop1 # Repeat loop

append\_second:

# Append msg2 to buffer

lb x5, 0(a2) # Load byte from msg2

beq x5, x0, end\_appach # If null terminator, end

sb x5, 0(a3) # Store byte in output buffer

addi a2, a2, 1 # Move to next character in msg2

addi a3, a3, 1 # Move to next position in output buffer

jal x0, append\_second # Repeat loop

end\_appach:

sb x0, 0(a3) # Null terminate the concatenated buffer

jalr x0, 0(x1) # Return from appach

lench:

addi a0, x0, 0 # Initialize length counter to 0

lench\_loop:

lb x5, 0(a1) # Load byte from the buffer

beq x5, x0, len\_end # If null terminator (0), end loop

addi a0, a0, 1 # Increment length counter

addi a1, a1, 1 # Move to next character

jal x0, lench\_loop # Repeat loop

len\_end:

jalr x0, 0(x1) # Return from lench