

Question 1:

```
; Author: Caitlin Coulombe, T00756521
; Date: June 8, 2024
; Course: COMP 2131
;
; I have a mac and am unable to run this because the mac uses arm64 architecture.
;
; Write a program in assembly language (using macros) to print out the following
messages on the screen
; Hello, programmers!
; Welcome to the world of,
; Linux assembly programming!

; data section
section .data
    msg1 db "Hello, programmers!", 0x0A, 0x00
    msg1_len equ $ - msg1      ; length of message 1
    msg2 db "Welcome to the world of,", 0x0A, 0x00
    msg2_len equ $ - msg2
    msg3 db "Linux assembly programming!", 0x0A, 0x00
    msg3_len equ $ - msg3

section .text
    global _start      ; this is the directive that makes the _start label visible to the
                        ; linker. It specifies the entry point of the program, which is where execution will
                        ; start.

_start:
    ; print msg1
    mov rax, 1          ; sys_write system call number (1 for x86_64)
    mov rdi, 1          ; file descriptor 1 is stdout
    mov rsi, msg1       ; pointer to the message
    mov rdx, msg1_len   ; length of the message
    syscall             ; calls the kernel

    ; print msg2
    mov rax, 1          ; sys_write system call number
    mov rdi, 1          ; file descriptor 1 is stdout
    mov rsi, msg2       ; pointer to the message
    mov rdx, msg2_len   ; length of the message
    syscall             ; calls the kernel
```

```
; print msg3
mov rax, 1          ; sys_write system call number
mov rdi, 1          ; file descriptor 1 is stdout
mov rsi, msg3       ; pointer to the message
mov rdx, msg3_len   ; length of the message
syscall             ; calls the kernel

; exit the application
mov rax, 60         ; sys_exit system call number (60 for x86_64)
xor rdi, rdi        ; exit status code 0 (successful termination)
syscall
```