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
```

Question 2

For loop

```
        .file   "sumWithFor.c"
        .text
.globl sumArguments
        .type   sumArguments, @function
sumArguments:
.LFB31:
        .cfi_startproc
        movl    %edi, %edx
        movl    $0, %eax
        cmpl    %esi, %edi
        jg      .L3
.L6:
        addl    %edx, %eax
        addl    $1, %edx
        cmpl    %edx, %esi
        jge     .L6
.L3:
        rep; ret
        .cfi_endproc
.LFE31:
        .size   sumArguments, .-sumArguments
        .section        .rodata.str1.8,"aMS",@progbits,1
        .align 8
.LC0:
        .string  "The sum of all the numbers between %d and %d, inclusive, is %d\n"
        .text
.globl main
        .type   main, @function
main:
```

```
.LFB30:
        .cfi_startproc
        subq    $8, %rsp
        .cfi_def_cfa_offset 16
        movl    $10, %esi
        movl    $5, %edi
        call    sumArguments
        movl    %eax, %ecx
        movl    $10, %edx
        movl    $5, %esi
        movl    $.LC0, %edi
        movl    $0, %eax
        call    printf
        movl    $0, %eax
        addq    $8, %rsp
        .cfi_def_cfa_offset 8
        ret
        .cfi_endproc
.LFE30:
        .size   main, .-main
        .ident  "GCC: (GNU) 4.4.7 20120313 (Red Hat 4.4.7-23)"
        .section        .note.GNU-stack,"",@progbits
```

While loop

```
        .file   "sumWithWhile.c"
        .text
.globl sumArguments
        .type   sumArguments, @function
sumArguments:
.LFB31:
        .cfi_startproc
        movl    $0, %eax
        cmpl    %esi, %edi
        jg      .L3
.L6:
        addl    %edi, %eax
        addl    $1, %edi
        cmpl    %edi, %esi
        jge     .L6
.L3:
        rep; ret
        .cfi_endproc
.LFE31:
        .size   sumArguments, .-sumArguments
```

```
        .section        .rodata.str1.8,"aMS",@progbits,1
        .align 8
.LC0:
        .string  "The sum of all the numbers between %d and %d, inclusive, is %d\n"
        .text
.globl main
        .type    main, @function
main:
.LFB30:
        .cfi_startproc
        subq    $8, %rsp
        .cfi_def_cfa_offset 16
        movl    $10, %esi
        movl    $5, %edi
        call    sumArguments
        movl    %eax, %ecx
        movl    $10, %edx
        movl    $5, %esi
        movl    $.LC0, %edi
        movl    $0, %eax
        call    printf
        movl    $0, %eax
        addq    $8, %rsp
        .cfi_def_cfa_offset 8
        ret
        .cfi_endproc
.LFE30:
        .size    main, .-main
        .ident   "GCC: (GNU) 4.4.7 20120313 (Red Hat 4.4.7-23)"
        .section        .note.GNU-stack,"",@progbits
```

Do … while loop

```
        .file    "sumWithDo.c"
        .text
.globl sumArguments
        .type    sumArguments, @function
sumArguments:
.LFB31:
        .cfi_startproc
        movl    $0, %eax
.L2:
        addl    %edi, %eax
        addl    $1, %edi
        cmpl    %esi, %edi
```

```
        jle     .L2
        rep; ret
        .cfi_endproc
.LFE31:
        .size   sumArguments, .-sumArguments
        .section        .rodata.str1.8,"aMS",@progbits,1
        .align 8
.LC0:
        .string  "The sum of all the numbers between %d and %d, inclusive, is %d\n"
        .text
.globl main
        .type   main, @function
main:
.LFB30:
        .cfi_startproc
        subq    $8, %rsp
        .cfi_def_cfa_offset 16
        movl    $10, %esi
        movl    $5, %edi
        call    sumArguments
        movl    %eax, %ecx
        movl    $10, %edx
        movl    $5, %esi
        movl    $.LC0, %edi
        movl    $0, %eax
        call    printf
        movl    $0, %eax
        addq    $8, %rsp
        .cfi_def_cfa_offset 8
        ret
        .cfi_endproc
.LFE30:
        .size   main, .-main
        .ident  "GCC: (GNU) 4.4.7 20120313 (Red Hat 4.4.7-23)"
        .section        .note.GNU-stack,"",@progbits
```

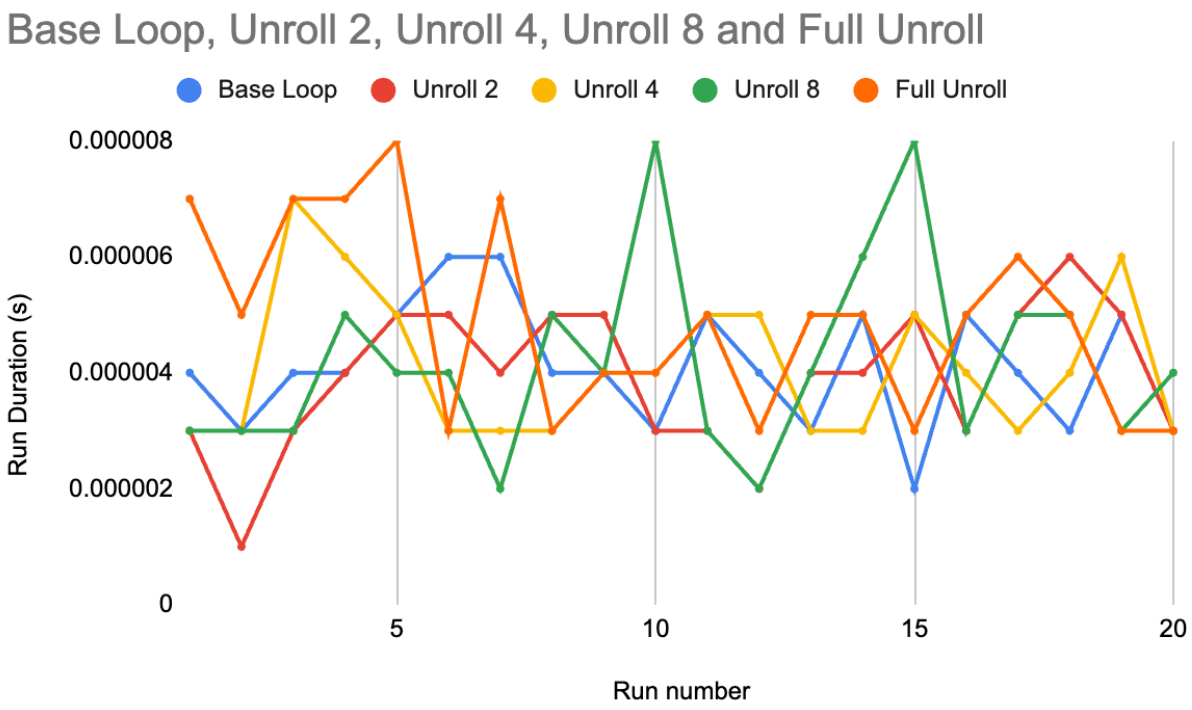
GoTo loop

```
        .file   "sumWithGoTo.c"
        .text
.globl sumArguments
        .type   sumArguments, @function
sumArguments:
.LFB31:
```

```
        .cfi_startproc
        movl    $0, %eax
        cmpl    %esi, %edi
        jg      .L5
.L3:
        addl    %edi, %eax
        addl    $1, %edi
.L4:
        cmpl    %edi, %esi
        jge     .L3
.L5:
        rep; ret
        .cfi_endproc
.LFE31:
        .size   sumArguments, .-sumArguments
        .section        .rodata.str1.8,"aMS",@progbits,1
        .align 8
.LC0:
        .string  "The sum of all the numbers between %d and %d, inclusive, is %d\n"
        .text
.globl main
        .type   main, @function
main:
.LFB30:
        .cfi_startproc
        subq    $8, %rsp
        .cfi_def_cfa_offset 16
        movl    $10, %esi
        movl    $5, %edi
        call    sumArguments
        movl    %eax, %ecx
        movl    $10, %edx
        movl    $5, %esi
        movl    $.LC0, %edi
        movl    $0, %eax
        call    printf
        movl    $0, %eax
        addq    $8, %rsp
        .cfi_def_cfa_offset 8
        ret
        .cfi_endproc
.LFE30:
        .size   main, .-main
        .ident  "GCC: (GNU) 4.4.7 20120313 (Red Hat 4.4.7-23)"
```

```
        .section        .note.GNU-stack,"",@progbits
```

Question 3



Average time for each version:

- Base loop: 0.0000041
- Unrolled by a factor of 2: 0.0000039
- Factor of 4: 0.0000041
- Factor of 8: 0.0000041
- Fully unrolled: 0.0000049