

CS 250 Lab 9 (Practice Test Problems)

Due April 10, 2024

Turn in your answers in some reasonable manner (on paper, in a file, etc). You can cut and paste the code into a compiler to check your work. All of the assembly programs here run without error, either in gcc or as.

Question 1: (10 points) What is the output of the following program?

```
.global main
.text
main:
    mov $12, %r12
    mov $6, %r13
    mov $24, %r14

loop:
    sub $-5, %r13
    add $-2, %r14
    mov $format, %rdi
    mov %r13, %rsi
    mov %r14, %rdx
    mov $0, %rax
    call printf
    cmp %r13, %r14
    jg loop
    ret
format:
    .asciz "%ld %ld\n"
```

Question 2: (10 points) What is the output of the following program?

```
.section .data
result_message:
    .asciz "The result is: %d\n"

.section .text
.global main
main:
    mov $42,%r8
    mov $12,%r9
    mov $17,%r10
    mov %r8,%rax
    mul %r10
    mov %rax,%r15
    mov %r15, %r8
loopstart:
    cmp $10000,%r8
    jge loopend
    mov %r8,%rax
    mul %r10
    mul %r9
    mov %rax,%r8

    jmp loopstart

loopend:
    mov %r15,%rax

    mov $result_message,%rdi
    mov %rax,%rsi
    mov $0,%rax
    call printf

    ret
```

Question 3: (10 points) Given the following x86-64 Linux assembly program, determine the exit code of the program when executed.

```
.section .text
.global _start

_start:
    movq $4, %rdi
    movq $1, %rax

factorial_loop:
    cmpq $1, %rdi
    jle end_factorial
    imulq %rdi, %rax
    decq %rdi
    jmp factorial_loop

end_factorial:
    movq %rax, %rdi
    movq $60, %rax
    syscall
```

Question 4: (10 points) What is the output of the following program?

```
.global main
.text
main:
    mov $10, %r14

up:
    mov $1, %rax
    mov $1, %rdi
    mov $format, %rsi
    mov $12, %rdx
    syscall

    cmp $12, %r14
    jl down
    ret

down:
    mov $12, %rdx
    syscall

    sub $2, %r14
    cmp $5, %r14
    jg up
    ret

.section .data
format:
    .ascii "Turn around\n"
```

Question 5: (10 points) What is the output of the following program?

```
.section .text
printf_string:
    .asciz "%lu\n"
times_ten:
    mov $10, %rax
    mul %rdi
    ret

.global main
main:
    mov $1, %r15
    mov %r15, %rdi
loop:
    call times_ten
    cmp $10000, %rax
    jg exit
    push %rax
    push %r15
    pop %rsi
    mov $printf_string, %rdi
    mov $0, %rax
    call printf
    pop %rdi
    jmp loop

exit:
    ret
```

Question 6: (10 points) You are given a task to multiply a number by two. The cpu you are using has faulty transistors, and the transistors used for the multiply instruction are not functioning. What other instruction can you use to multiply a number by two?