

6) What is the value of `%eax` after the following assembly code: (10 points)

```
    movl    $37, %eax
    cmpl   %eax, $50
    jle    .L1
    jmp    .L2
.L1:
    movl   $1, %eax
.L2:
```

7) Convert the following C code to assembly. Assume `x` is stored in `%edx`. (18 points)

```
    if(x == 10) {
        x = 0;
    } else {
        x++;
    }
}
```

8) Convert the following C code to assembly. Assume `x` is stored in `%edx` and `r` is stored in `%eax`. (18 points)

```
    int x;
    int r = 0;
    for(x=0; x < 20; x++) {
        r += x;
    }
}
```

- 9) Write equivalent C code for the assembly code below. You'll need to make up some variable names. (20 points)

```
.globl main
main:
    pushl    %ebp
    movl    %esp, %ebp
    subl    $8, %esp
    andl    $-16, %esp
    movl    $0, %eax
    subl    %eax, %esp
    movl    $0, -4(%ebp)
.L2:
    cmpl    $9, -4(%ebp)
    jle    .L4
    jmp    .L3
.L4:
    leal    -4(%ebp), %eax
    incl    (%eax)
    jmp    .L2
.L3:
    leave
    ret
```

- 10) What are the two important things always stored in the previous stack frame? At what offset from `%ebp` are they found? (8 points)

11) Write assembly code for the following C function: (18 points)

```
int mystery(int x, int y) {  
    int result = x*2;  
    result += y;  
    return result;  
}
```

12) Write an equivalent C function for the following assembly function. (15 points)

```
.globl g  
g:  
    pushl    %ebp  
    movl    %esp, %ebp  
    movl    8(%ebp), %eax  
    imull   8(%ebp), %eax  
    addl    8(%ebp), %eax  
    incl    %eax  
    popl    %ebp  
    ret
```

13) Describe the purpose of the call, leave, and ret instructions. What instructions can they be replaced with? (12 points)

14) What would be the output if this assembly code was assembled and ran? Why? (15 points)

```
.LC0:
    .string "%d %d %d %d\n"
.globl main
main:
    pushl    %ebp
    movl    %esp, %ebp
    subl    $40, %esp
    andl    $-16, %esp
    movl    $0, %eax
    subl    %eax, %esp
    movl    $0, 24(%esp)
    movl    $1, 20(%esp)
    movl    $2, 16(%esp)
    movl    $3, 12(%esp)
    movl    $4, 8(%esp)
    movl    $5, 4(%esp)
    movl    $.LC0, (%esp)
    call    printf
    leave
    ret
```

15) Define caller save and callee save registers. Of the six general-purpose registers, which are caller save and which are callee save? (10 points)

16) Draw the entire stack at its largest (before it starts contracting) after main calls g(8). What is the return value for g(8)? (18 points)

```
.globl g
g:
    pushl    %ebp
    movl    %esp, %ebp
    subl    $8, %esp
    cmpl    $10, 8(%ebp)
    jne     .L2
    movl    $1, -4(%ebp)
    jmp     .L1
.L2:
    movl    8(%ebp), %eax
    incl    %eax
    movl    %eax, (%esp)
    call   g
    addl    $2, %eax
    movl    %eax, -4(%ebp)
.L1:
    movl    -4(%ebp), %eax
    leave
    ret
```