CSE 351 WI 12 Midterm Solutions

Question 1: No solution provided. Concepts are similar to question 2.

Question 2:

```
fib:
pushq %rbp
pushq %rbx
subq $8, %rsp
movl %edi, %ebx
movl %edi, %eax
cmpl $1, %edi
jle return
leal -1(%rdi), %edi
call fib
movl %eax, %ebp
leal -2(%rbx), %edi
call fib
addl %ebp, %eax

return:
addq $8, %rsp
popq %rbx
popq %rbp
ret
```

*note: Even though we are writing for x86_64, sometimes it is unavoidable to use the stack to save registers or other values.

Question 3:
(a) 0000 0000 0000 0011
(b) 0000 0000 0000 1110
(c) 1111 1111 1111 0010
(d) 1111 1111 1111 0101

Question 4:
(a) 0x1006
(b) Register that callee is responsible for saving in its stack frame before using. Callee must restore register before returning to caller.

(c) 0xC0500000 = 0b 1100 000 0010 1000 0000 0000 0000
s = 1, exp = 0b1000 0000, frac = 101 0000 0000 0000 0000 0000

For 32-bit floating point: 1 bit sign, 8 bits exponent, 23 bits fraction
$V = (-1)^S \times M \times 2^E$
$E = exp - bias$, $bias = 2^{k-1} - k$, k = number of exponent bits (8)
$M = 1 + frac$, $1 <= M < 2$
Strategy: sign is 1 if negative, 0 if positive. Then, divide 3.25 by 2 some number of times (E) until satisfies $1 <= M < 2$. Here, dividing 3.25 by 2 once (E = 1) satisfies this and gives M = 1.625. Thus, exp = E + bias = 1 + 127 = 128 = 0b1000 0000. Since M = 1.625, frac = 0.625 = 1/2 + 1/8, giving frac = 0b101 0000 0000 0000 0000 0000. Remember, frac represents the negative powers of 2.

*note: do not worry about how denormalized or special floating point values are stored.

(d) FALSE (first 6 passed in registers)
(e) Little (LSB stored at lowest address)
(f) TRUE (Read about casting between unsigned and signed, and type promotion during comparisons)
(g) -32768