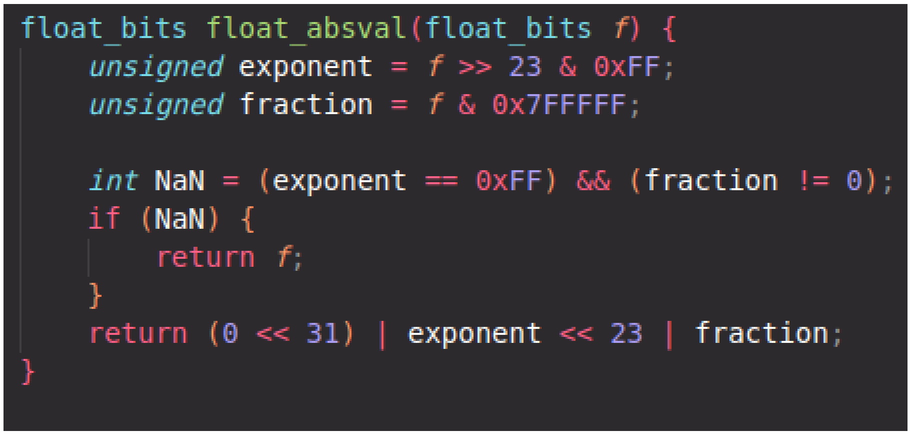
Noah Hanks

ECEN 424 HW 2

2.87

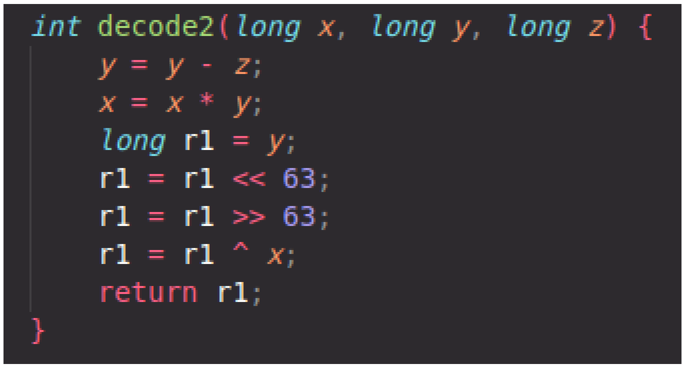
Description	Hex	*M*	*E*	*V*	*D*
-0	BC00	0	-14	-0	-0.0
Smallest value > 2	4001	1025/1024	1	1025/512	2.00195
512	6000	1	9	512	512.0
Largest denormalized	03FF	1023/1024	-14	1025/1024 * 2^-14	0.000061
-∞	FC00	--	--	-∞	-∞
Number with hex representation 3BB0	3BB0	123/64	-1	123/128	0.96093

2.93



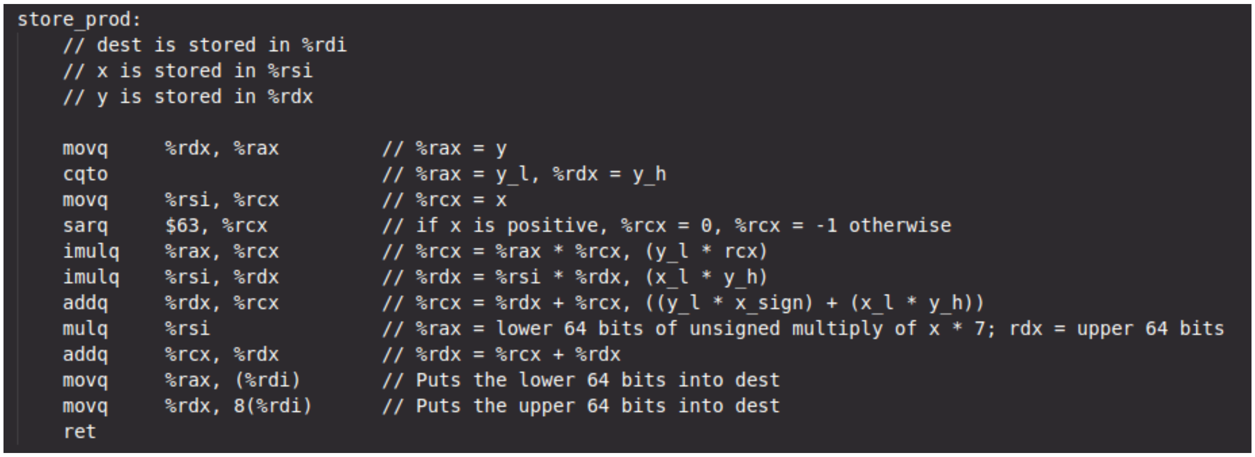
```
float_bits float_absval(float_bits f) {
    unsigned exponent = f >> 23 & 0xFF;
    unsigned fraction = f & 0x7FFFFF;

    int NaN = (exponent == 0xFF) && (fraction != 0);
    if (NaN) {
        return f;
    }
    return (0 << 31) | exponent << 23 | fraction;
}
```

3.58



```c
int decode2(long x, long y, long z) {
    y = y - z;
    x = x * y;
    long r1 = y;
    r1 = r1 << 63;
    r1 = r1 >> 63;
    r1 = r1 ^ x;
    return r1;
}
```

3.59



```
store_prod:
    // dest is stored in %rdi
    // x is stored in %rsi
    // y is stored in %rdx

    movq     %rdx, %rax         // %rax = y
    cqto                        // %rax = y_l, %rdx = y_h
    movq     %rsi, %rcx         // %rcx = x
    sarq     $63, %rcx          // if x is positive, %rcx = 0, %rcx = -1 otherwise
    imulq    %rax, %rcx         // %rcx = %rax * %rcx, (y_l * rcx)
    imulq    %rsi, %rdx         // %rdx = %rsi * %rdx, (x_l * y_h)
    addq     %rdx, %rcx         // %rcx = %rdx + %rcx, ((y_l * x_sign) + (x_l * y_h))
    mulq     %rsi               // %rax = lower 64 bits of unsigned multiply of x * 7; rdx = upper 64 bits
    addq     %rcx, %rdx         // %rdx = %rcx + %rdx
    movq     %rax, (%rdi)       // Puts the lower 64 bits into dest
    movq     %rdx, 8(%rdi)      // Puts the upper 64 bits into dest
    ret
```

3.60

A) x = %rdi
n = %ecx
result = %rax
mask = %rdx

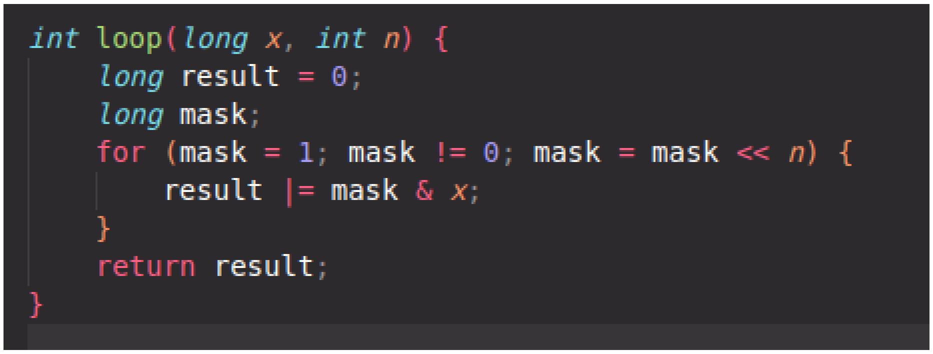
B) result = 0
mask = 1

C) The loop continues for as long as mask != 0

D) The mask gets updated by shifting its bits to the left n times

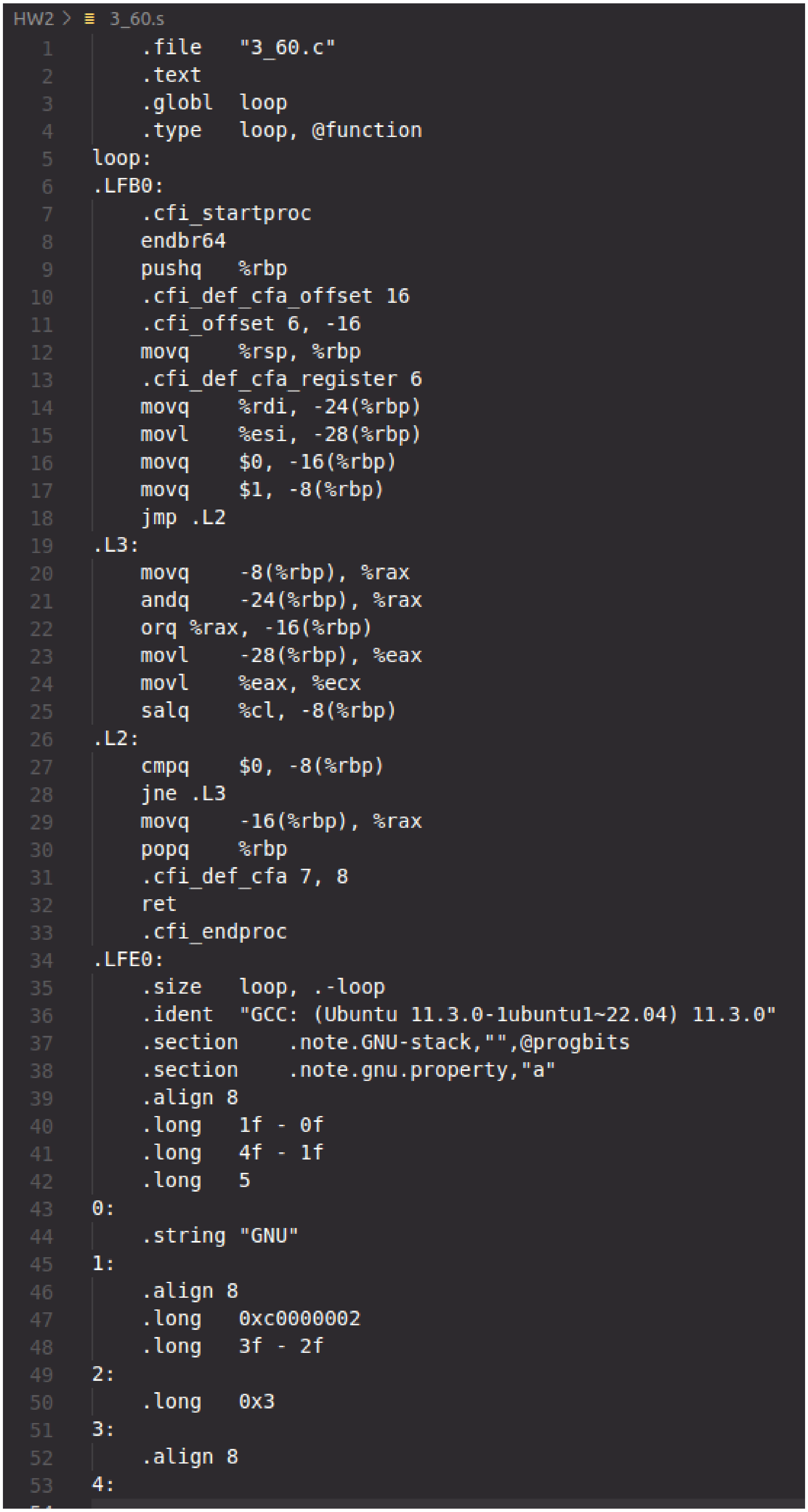
E) The result gets updated by result |= (mask & x)

F)



```c
int loop(long x, int n) {
    long result = 0;
    long mask;
    for (mask = 1; mask != 0; mask = mask << n) {
        result |= mask & x;
    }
    return result;
}
```

Compiled using “gcc -S -m64 3_60.c”



```
	.file	"3_60.c"
	.text
	.globl	loop
	.type	loop, @function
loop:
.LFB0:
	.cfi_startproc
	endbr64
	pushq	%rbp
	.cfi_def_cfa_offset 16
	.cfi_offset 6, -16
	movq	%rsp, %rbp
	.cfi_def_cfa_register 6
	movq	%rdi, -24(%rbp)
	movl	%esi, -28(%rbp)
	movq	$0, -16(%rbp)
	movq	$1, -8(%rbp)
	jmp	.L2
.L3:
	movq	-8(%rbp), %rax
	andq	-24(%rbp), %rax
	orq	%rax, -16(%rbp)
	movl	-28(%rbp), %eax
	movl	%eax, %ecx
	salq	%cl, -8(%rbp)
.L2:
	cmpq	$0, -8(%rbp)
	jne	.L3
	movq	-16(%rbp), %rax
	popq	%rbp
	.cfi_def_cfa 7, 8
	ret
	.cfi_endproc
.LFE0:
	.size	loop, .-loop
	.ident	"GCC: (Ubuntu 11.3.0-1ubuntu1~22.04) 11.3.0"
	.section	.note.GNU-stack,"",@progbits
	.section	.note.gnu.property,"a"
	.align 8
	.long	1f - 0f
	.long	4f - 1f
	.long	5
0:
	.string	"GNU"
1:
	.align 8
	.long	0xc0000002
	.long	3f - 2f
2:
	.long	0x3
3:
	.align 8
4:
```