PART 2: Evaluating Execution Time

A) Finding the F(6) for both loop and recursive functions.

Loop Function(Non-recursive)

Category	Instructions	CPI(Clock Cycle per Instruction)	Instruction Count
A: Arithmetic and Comparison	add, addu, sub, slt, etc..	1	36
B: Memory	lw, sw	8	32
D: Branch and Jump	beq, bnq, j, jal	2	6

Recursive Function

Category	Instructions	CPI(Clock Cycle per Instruction)	Instruction Count
A: Arithmetic and Comparison	add, addu, sub, slt, etc..	1	111
B: Memory	lw, sw	8	286
D: Branch and Jump	beq, bnq, j, jal	2	25

B) Total clock cycles for each function.

Loop Function(Non-recursive)

36*1 + 32*8 + 6*2 = 304 clock cycles

Recursive Function

111*1 + 286*8 + 25*2 = 2449 clock cycle

C) How much faster would the functions be if a better data cache reduced the average memory operations time to 4 cycles?

Loop Function(Non-recursive)

36*1 + 32*4 + 6*2 = 176 clock cycles

Loop function would be 1.72 times faster if a better data cache reduced the average memory operations time to 4 cycles.

Recursive Function

111*1 + 286*4 + 25*2 = 1305 clock cycles

Loop function would be 1.87 times faster if a better data cache reduced the average memory operations time to 4 cycles.

PART 3: Representing Numbers

A)

Maximum integer value = 2147483647_{ten}

Two's complement: $01111111\ 11111111\ 11111111\ 11111111_{two}$

Hexadecimal form: $7FFFFFFF_{hex}$

Minimum integer value = -2147483648_{ten}

Two's complement: 10000000 00000000 00000000 00000000

Hexadecimal form: 80000000_{hex}

B)

32 bit floating point representation of -22.2_{ten}

$11000001\ 10110001\ 10011001\ 10011010_{two}$