1. Translate the following decimal number into its binary representation.

1.25	1.01
2.75	10.11
15.5	1111.1
13.625	1101.101
50.125	110010.001

2. Write the value that is returned of the function "int foo()", according to the x86 assembly codes.

```
foo:
        pushq   %rbp
        movq    %rsp, %rbp
        movl    $3, -16(%rbp)
        movl    $4, -12(%rbp)
        movl    $0, -8(%rbp)
        movl    $1, -4(%rbp)
        jmp     .L2
.L3:
        movl    -4(%rbp), %eax
        imull   -16(%rbp), %eax
        movl    %eax, -4(%rbp)
        addl    $1, -8(%rbp)
.L2:
        movl    -8(%rbp), %eax
        cmpl    -12(%rbp), %eax
        jl      .L3
        movl    -4(%rbp), %eax
        leave
        ret
```

Returned value=81

3. Write the byte codes of the following Y86 instructions.

Y86 instructions	Byte codes
rrmovl %ebx,%ecx	2031
jmp 0xabc	70bc0a0000
addl %ebx,%eax	6030
call 0x1234	8034120000
rmmovl %ecx,0x40(%ebx)	401340000000
je 0x280	7380020000
pushl %eax	a00f(或 a008)

4. We now have a piece of Y86 code, tell the value of %eax when the program ends; and give a description about what the program has done.

```
0x000:
0x000: 308400020000    irmovl 0x200, %esp
0x006: 308500020000    irmovl 0x200, %ebp
0x00c: 7058000000      jmp 0x58

0x011:
0x011: a058            pushl %ebp
0x013: 2045            rrmovl %esp,%ebp
0x015: a008            pushl %eax
0x017: a018            pushl %ecx
0x019: 308102000000    irmovl 0x2, %ecx
0x01f: 6110            subl %ecx, %eax
0x021: 7631000000      jg 0x31
0x026: 308001000000    irmovl 0x1, %eax
0x02c: 7051000000      jmp 0x51

0x031:
0x031: 8011000000      call 0x11
0x036: a028            pushl %edx
0x038: 2002            rrmovl %eax, %edx
0x03a: 5005fcffffff    mrmovl -4(%ebp), %eax
0x040: 308101000000    irmovl 0x1, %ecx
0x046: 6110            subl %ecx, %eax
0x048: 8011000000      call 0x11
0x04d: 6020            addl %edx, %eax
0x04f: b028            popl %edx
0x051:
0x051: b018            popl %ecx
0x053: b058            popl %ebp
0x055: b058            popl %ebp
0x057: 90              ret
0x058:
0x058: 308007000000    irmovl 0x7, %eax
0x05e: 8011000000      call 0x11
0x063: 10              nop

0x200:
0x200:
```

%eax: _13__

description: _ Fibonacci__