※ 考生請注意：本試題不可使用計算機
(請字體清晰,並依次序作答, 否則不予計分)

1. With digital electronics, we represent a number as a string of bits. (20%)

a. Given a positive integer n, how many binary digits are needed to represent n?

b. High-dynamic-range numbers are often represented in floating-point format. A floating-point number has two components; the exponent E and the fraction. The value represented by a floating point number is given by $(-1)^S$ (1+Fraction) 2^E, where S is the sign of the number, E is the value of the exponent field. Suppose that all fraction bits are significant, how many significant digits in decimal number system when we use 23 binary digits to store the fraction.

2. Each real number either is an integer itself or sits between two consecutive integers: For each real number x, there exists a unique integer n such that $n \leqq x < n+1$. The floor of a number is the integer immediately to its left on the number line. Plot the graph of the floor function. (10%)

3. Prove that if f(x) is O(h(x)) and g(x) is O(k(x)), then f(x)+g(x) is O(G(x)), where, for each x in the domain, G(x)=max(|h(x)|,|k(x)|). (20%)

4. Assume n is a positive integer and consider the following algorithm segment: (10%)

```
s:=0
for i:=1 to n
    for j:=1 to i
        s:=s+j(i-j+1)
    next j
next i
```

a. Compute the actual number of additions, subtractions, and multiplications that must be performed when this algorithm segment is executed.

b. Use the theorem on polynomial orders to find an order for this algorithm segment.

（背面仍有題目，請繼續作答）

※ 考生請注意：本試題不可使用計算機

5. Define a sequence $a_1, a_2, a_3, \ldots$, recursively as follows: (10%)

$a_1 = 1$

$a_k = 2a_{\lfloor k/2 \rfloor}$, for all integers $k \geqq 2$.

Use iteration to guess an explicit formula for this sequence.

6. The game of poker is played with an ordinary deck of cards containing 52 cards divided into four suits. How many five-card poker hands contain two pairs? (10%)

7. (Let A be a finite-state automaton with next-state function N. Given any states s and t in A, where

- s is 0-equivalent to t <=> [either s and t are both accepting states or they are both non accepting states]
- for every integer $k \geqq 1$, s is k-equivalent to t <=> [s and t are (k-1)-equivalent, and for any input symbol m, N(s, m) and N(t, m) are also (k-1)-equivalent.

Find the 0-equivalence classes, the 1-equivalence classes, and the 2-equivalence classes for the states of the automaton shown below. (20%)