※ 考生請注意：本試題 □可 ☑不可 使用計算機

1. Answer the following questions briefly: (40%)

(1) Suppose that n ($n > 1$) people are positioned so that each has a unique nearest neighbor and each person has a pie that is to throw at and definitely hit the nearest neighbor. A *survivor* is defined as a person that is not hit by a pie. Prove or disprove that if n is odd, one of two persons farthest apart is a *survivor*. (5%)

(2) Let L be the set of all strings, including the null string λ, which can be generated by repeated application of the following rules:

(i) if α belongs to L, then aαb belongs to L and bαa belongs to L.

(ii) if α belongs to L and β belongs to L, then $\alpha\beta$ belongs to L.

For example, ab is in L, for if we take $\alpha = \lambda$ (null string), then α belongs to L and the first rule states that ab = aαb belongs L. Similarly, ba belongs to L. As another example, aabb is in L, for if we take α = ab, then α belongs to L and the first rule states that aabb = aαb belongs to L.

Is baabab in L? (5%)

(3) Give an argument using rules of inference to show if the conclusion follows from the hypotheses.

Hypotheses: Everyone in the class has a graphing calculator. Everyone who has a graphing calculator understands the trigonometric functions.

Conclusion: Ralphie, who is in the class, understands the trigonometric functions. (5%)

(4)How many integer solutions of $x_1 + x_2 + x_3 + x_4 = 17$ satisfy $x_1 \geq 0$, $x_2 \geq 1$, $x_3 \geq 2$, $x_4 \geq 3$?(5%)

(5) We have two fair dice. Suppose that two dice are rolled and we are told that the outcome of at least one die is 5. What is the probability of getting the sum of the values of the outcomes that add to exactly 10? (5%)

(6) Define the truth table for *op* as follows. Prove or disprove that

$a \text{ op } b \equiv b \text{ op } a$. (5%)

a	b	a op b
T	T	T
T	F	F
F	T	F
F	F	T

(7) Design a full-adder circuit that uses two half adders and one OR gate (5%).

(Hint: Use a half adder as a black box. You do not need to design a half adder.)

(背面仍有題目,請繼續作答)

※ 考生請注意：本試題 □可 ☑不可 使用計算機

(8) How many times is the print statement executed?

```
for  i1 := 1 to n do
  for i2 := 1 to i1 do
    for i3 := 1 to i2 do
      ...
        for ik :=1 to ik-1 do
            Print i1, i2, ..., ik.
```

Note that each line of output consists of k integers: $i_1 i_2 \ldots i_k$ where $n \geq i_1 \geq i_2 \geq \ldots \geq i_k \geq 1$ (condition 1) and that each sequence $i_1 i_2 \ldots i_k$ satisfying condition 1 occurs. (5%)

2. A relation R is defined on the set of eight-bit strings by $s_1 R s_2$ if the first four bits of s_1 and s_2 are the same. Prove or disprove: R is an equivalence relation. (10%)

3. Consider a version of the game of *nim*. Initially, there are n piles, each containing a number of identical coins. There are two players and players take alternate moves. A move consists of removing one or more coins from any one pile. The player who removes the last coin loses. Suppose there are two piles: one has three coins and the other one contains two coins. You are to move first. Is there any move that guarantees you will win? Why or why not? (15%)

4. Consider the following algorithm that evaluates a polynomial:

Input: The sequence of coefficients $c_0, c_1, \ldots, c_n$, the value t and n, where the coefficients and n are non-negative integers.

Output: poly(t)

```
procedure poly(c,n,t) //hint: Imagine poly(c,n,t) as given cn, n and t,
  if n = 0 then     // and poly(c,n-1, t) as given cn-1 ,n-1 and t.
    return c0       //That is, c is the array for the coefficients and n is c's index.
  return (t* poly(c, n-1, t) + cn)
end poly
```

Let b_n be the number of multiplications required to compute $p(t)$ (Note: the multiplication is denoted as a "*" in the algorithm.)(20%)

(i) Find a recurrence relation and an initial condition for b_n (10%)

(ii) Compute b_1, b_2 and b_3(5%)

(iii)Solve the recurrence relation of (i). (5%)

5. Write a program that determines whether a graph contains an Euler cycle. (15%)

(i)Write your algorithm first. (5%)

(ii)Write your program. (10%)

(Note: You are not required to find an Euler cycle of the graph. Use a good programming style to make your program readable. Use C-family language (such as C, C++, C#), or Java.)