1. The following Boolean algebra expression is given as:

$$Q = \overline{A}(\overline{B}C + BC + B\overline{C}) + ABC$$

   a) Convert this logical equation into an equivalent SOP (sum of product) term and use a truth table to show all the possible combinations of input conditions that will produces an output. (5%)
   b) Draw a logic gate diagram for the expression (5%)

2. The format of IEEE single-precision floating-point standard representation requires 23 fraction bits, 8 exponent bits, and 1 sign bit, with a total of 32 bits for each word. Convert the following binary number in the IEEE single-precision format to its decimal format. (10%)

1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3. For the following number list, perform a bubble sort in ascending order and show the list after each iteration. How many comparisons and swaps of elements are made when the list is sorted? (10%)

4	1	3	2	5

4. If we have a depth-first traversal of a tree's pre-order and post-order as following:

   pre-order: A, B, D, C, E, G, F, H, I

   post-order: D, B, G, E, H, I, F, C, A

   Please draw the corresponding binary tree. (10%)

5. In RSA algorithm, suppose p is 7, q is 11, and e is 13, Find the n and d. (10%)

6. Given an initially empty hash table with capacity 11 and hash function H(x) = x mod 11. Keys 0, 1, 8, 9 are already in the table. Insert keys 52, 44, 56, 53, 61, 64 using linear probing (in that order). (10%)

0	1	2	3	4	5	6	7	8	9	10
0	1							8	9	

7. Let G be the grammar shown below. The ε (epsilon) stands for the empty string. Draw the parse tree for the string w = xx$ (10%)

S → A$
A → (AB)
A → ε
B → (A)
B → x

見背面

8. Construct the even parity Hamming code word for a data byte 1001101. (10%)

9. Suppose you run the following assembly program on a computer. The computer consists of 8 memory registers (**M[1]** to **M[8]**), 2 input registers (**I[A]** and **I[B]**), and one output register (**OUT**). Consider the following operations and controls for the assembly.

```
12 MOV I[A] M[1]
13 MOV I[B] M[2]
14 ADD I[A] I[B]
15 MOV I[A] OUT
16 MOV I[B] M[3]
17 ADD I[A] I[B]
18 MOV M[4] OUT
19 HALT
```

OPERATIONS AND CONTROLS
1: MOVE X Y: X = Y
2: ADD X Y: OUT = X + Y
3: SUB X Y: OUT = X - Y
4: HALT: halt

Suppose the initial contents of the registers are: **M[1]** = 0001; **M[2]** = 0011; **M[3]** = 0011; **M[4]** = 0100. Answer the questions below. Write your answers in the format of 4-digit decimal numbers.

a) What is the value of **OUT** after the instruction at location 15 completes? (3%)

b) What is the value of **OUT** when the program halts? (2%)

c) What is the value of **M[4]** after the instruction at location 16 completes? (2%)

d) What is the value of **M[4]** when the program halts? (3%)

10. A happy number is defined by the following process. Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers. In other words, numbers that are happy follow a sequence that ends in 1. All unhappy numbers follow sequences that reach the cycle. For example, 19 is a happy number, as the associated sequence is:

$$1^2 + 9^2 = 82$$
$$8^2 + 2^2 = 68$$
$$6^2 + 8^2 = 100$$
$$1^2 + 0^2 + 0^2 = 1.$$

Design a computer program to find happy numbers up to 1000. You may use any computer programming language for this problem. (10%)

試題隨卷繳回