1. Single choice (with single answer) problems. (4% each; 20% in total)

   a) Given the following block of C program code. Which line contains an error?

i	int m = 5;
ii	double n = 99;
iii	double o = m;
iv	int p = n;

   b) Which of the following is not executed by the BIOS?
      i. Memory test
      ii. Load application software
      iii. Print a test page on the printer
      iv. Load the operating system

   c) Which of the following is NOT an internet protocol?
      i. SMTP
      ii. FTP
      iii. HTML
      iv. HTTP

   d) Which of the following statement is not true?
      i. Both Turing or von Neumann models has the concept of the program
      ii. In the Turing model, the output data depends only on the input data without the program
      iii. A universal Turing machine can do any computation if the appropriate program is provided
      iv. A program in the von Neumann model is made of a finite number of instructions

   e) In which of the following addition problems (using two's complement notation) does an overflow error occur?
      i. 1100 + 1100
      ii. 1100 + 0100
      iii. 0011 + 1010
      iv. 0100 + 0100

2. For each row below containing a binary, decimal, and hexadecimal number. Answer the largest value in a row. Provide the number type (binary, decimal, and hexadecimal) in your answer. Answer "All Equal" if all three entries have the same value. (10%)

	Binary	Decimal	Hexadecimal	All equal
(a)	10	2	2	*All Equal*
(b)	1100	12	C	*All Equal*
(c)	10010	18	12	*All Equal*
(d)	100000	34	19	*All Equal*
(e)	1111110	250	FE	*All Equal*

3. If a 5-minute song is sampled at 44 kHz, and each sample is stored as two 8-bit numbers. How much space in Megabyte it needs to store the song? (10%)

4. Assume that a system that is similar to the IEEE standard. The system uses 8 bits to represent the floating-point. The leftmost bit is the sign bit. The following 3 bits are exponent stored in Excess 3 system. The final 5 bits store the mantissa after normalization. Decode the following bit patterns into a floating point: (a) 01011001 and (b) 10101100. (10%)

見 背 面
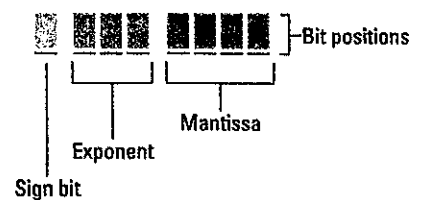
5. Suppose there are three input, *in1*, *in2*, and *in3*. Draw a digital circuit that implements the logical expression: (10%)

    not (not *in1* and *in2*) or ((not *in2* and *in3*) or *in3*)

6. Show the values of registers R0, R1, R2, R3, and R4 after the each of the instructions in the assembly code below. (20%)

1	MOV R4, #2
2	MUL R0, R4, R4
3	ADD R1, R4, #1
4	SUB R2, R4, #1
	ADD R3, R0, R4

7. The follow is a function to raise grades written in C language. The function calls another function named "sort" to sort grades in ascending order. Identify each buggy line in the program. Explain the bugs concisely. You may either explain the bug without fixing it, or provide a snippet of code that would correct it. You do not need to write complete sentences or complete corrected lines of code. (20%)

```
1   public void inflateGrades(int[] studentGrades) {
2       int[] copyOfGrades = int[studentGrades.length];
3
4       // copy the contents of studentGrades 1 at a time
5       copyOfGrades = studentGrades;
6
7       copyOfGrades = sort(copyOfGrades);
8
9       // for each grade in student grades...
10      for (int i = 0; i <= studentGrades.length; i++) {
11          // if student has maximum grade, give 100
12          if (studentGrades[i] == copyOfGrades[0]) {
13              studentGrades[i] = 100;
14              break;
15          }
16
17          // otherwise find the next grade after toBeFound
18          int toBeFound = studentGrades[j];
19          for (int j = 0; j <= studentGrades.length; j++) {
20              if (toBeFound == copyOfGrades[j]) {
21                  while(toBeFound == copyOfGrades[j])
22                      j--;
23              }
24              studentGrades[j] = copyOfGrades[j];
25          }
26      }
27   }
28  }
```

```
public static void sort(int[] array) {
    for (int i = 0; i < array.length - 1; i++) {
        int a = array[i];
        int b = i;
        for (int j = i + 1; j < array.length; j++) {
            if (array[j] < a) {
                a = array[j];
                b = j;
            }
        }
        int temp = array[i];
        array[i] = array[b];
        array[b] = temp;
        printArray(array);
    }
}
```