國 立 臺 北 科 技 大 學

九十六學年度電資碩士在職專班招生考試

計算機概論 試題

填 准 考 證 號 碼　　第一頁　共一頁

注意事項：

1. 本試題共【9】題，配分共 100 分。
2. 請按順序標明題號（大題及子題）作答，不必抄題。
3. 全部答案均須答在答案卷之答案欄內，否則不予計分。
4. 請掌握時間，回答問題務必切題、簡潔、精確，非必要的文句對得分並無幫助。

1. (a) A list is stored as an *ordered binary tree* for applying binary search algorithm. Draw the *ordered binary tree* for the list of letters B, E, G, H, J, K, N and P. Your choice of the root should be able to make the tree as balance as possible. (5%)
   (b) Redraw the tree if a new letter M is added to the list in (a). (3%)
   (c) Indicate the path traversed by the *binary search algorithm* when applied to the tree in (b) in searching for the letter J. (2%)

2. Explain the following terms precisely and concisely. The full (English) name must be provided in your answer if a term is given in its abbreviated form. (12%)
   (a) "CAM" in memory technology (b) RISC (c) Extranet (d) Proxy server

3. Answer the following questions:
   (a) What is the meaning of *process starvation*? (5%)
   (b) What is the meaning of *polymorphism* in OOP? (5%)

4. (a) What are the differences between *virus* and *spyware*? (4%)
   (b) What does the *spam* mean? (4%)
   (c) What are the differences between *iterative structure* and *recursive structure* in programming languages? (4%)

5. (a) Show how the array in Figure 1 would be arranged in main memory when stored in *row-major order*. Assume that the data are stored in the memory from high location (say,

x) to low location orderly. (5%)

5	4	1
3	2	9
7	8	6

Figure 1 A 3x3 two-dimensional array.

(b) Similar to (a), give a *formula* for finding the entry in the *i*th row and *j*th column of an *nxm* two-dimensional array if it is stored in *column-major order* in the memory. (5%)

6. What is the three-step process in a *machine cycle*? Explain each step. (8%)

7. Compare the following terms from various aspects:
   (a) *IPv4* versus *IPv6*? (5%)
   (b) *TCP* versus *UDP*? (5%)
   (c) *Repeater* versus *Switch*? (6%)

8. Design a *BCD adder/subtracter* in which the circuit has inputs of two BCD numbers and one *control*(*ctrl*). When *ctrl* = 0 (or 1), the circuit will perform BCD addition: *BCD#1 + BCD#2* (or BCD subtraction: *BCD#1 - BCD#2*). Ignore the carry/borrow to the BCD adder/subtracter. You can use a 4-bit binary full adder as a building block (Figure 3). Draw the *logic circuit* of your design. (10%)

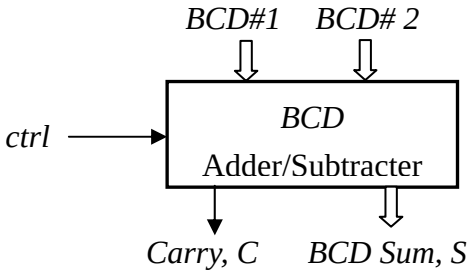

Figure 2 A BCD adder/subtracter to be designed.

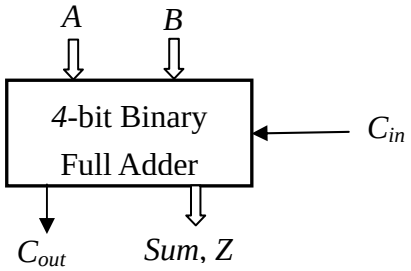

Figure 3 A 4-bit binary full adder.

9. (a) $(3A5.AA)_{11} = (\ ?\ )_8$. (4%)
   (b) 110000 – 110101 = ___?___, using *unsigned* representation. (4%)
   (c) Convert $(78)_{11}$ to “6-4-1-1” code. (4%)