國立臺灣科技大學 111 學年度碩士班招生試題

系所組別：電子工程系碩士班甲組

科　　目：資料結構

**( 總分為 100 分；所有試題務必於答案卷內頁依序作答，否則不予計分 )**

1. (15%)
   (1) Please explain f(n)=O(g(n)), where O is the "big oh" notation. (5%)
   (2) rsum() is a recursive summing function, please analyze the time complexity by the "big oh" notation. (10%)

```
float rsum(float list[], int n)
{
    if(n)
        return rsum(list,n-1)+list[n-1];
    return list[0];
}
```

2. (10%) Please write strcat(char *dest, chat *src) function to concatenate dest and src strings, and return result in dest. For example, assume that s1 string is "Hello " and s2 string is "World", after calling strcat(s1, s2), s1 will become "Hello World".
3. (15%) A queue is an array in which insertions and deletions take place at different ends. The end index at which a new element is added is rear, and another end index at which an element is deleted is front. Assume that addq() and deleteq() are two functions to add a new element to a queue and delete an element from a queue. Please rewrite addq() and deleteq() if a circular queue is used.

```
void addq(element item) {
    if (rear==MAX_QUEUE_SIZE)
        queueFull(); // print an error message
    else
        queue[rear++]=item;
}
element deleteq(){
    if(front==rear)
        return queueEmpty(); // return an error key
    return queue[front++];
}
```

4. (10%) Assume that a linked list has the following definition (i.e., struct listNode) and please write delete(struct listNode *first, struct listNode *trail, struct listNode *x) function to delete x from the linked list, where trail is the preceding node of x and first is the front of the linked list.

```
struct listNode {
     int data;
     struct listNode *next; // point to the next node
};
```

國立臺灣科技大學 111 學年度碩士班招生試題

系所組別：電子工程系碩士班甲組

科　　目：資料結構

**( 總分為 100 分；所有試題務必於答案卷內頁依序作答，否則不予計分 )**

5. (15%) Consider the sequence $L$: 5,55,4,44,3,33,2,22,1,11. Answer the following question:

   (1) Write the status of the sequence $L$ after each phase of the insertion sort. (5%)

   (2) Write the status of the sequence $L$ after each phase of the radix sort with radix = 10. (5%)

   (3) What is stable sorting? Please give a brief example to explain. (5%)

6. (25%) Consider the graph $G$.

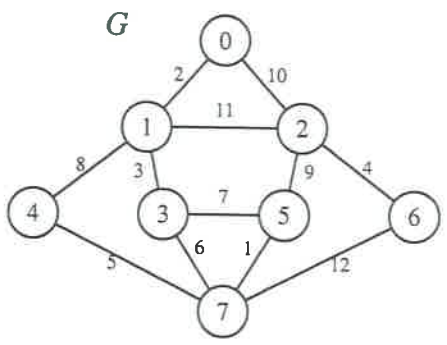
   (1) Show its adjacency matrix. (5%)

   (2) Show its adjacency list. (5%)

   (3) Show the depth-first spanning tree starting from node 0 based on your adjacency list in (2). (5%)

   (4) Show the breadth-first spanning tree starting from node 0 based on your adjacency list in (2). (5%)

   (5) Find a minimum spanning tree using Kruskal's algorithm. (5%)




7. (10%) Write an algorithm for finding the maximum number in a binary search tree.