Dept. of Computer Science and Engineering, undergraduate
National Sun Yat-sen University
Data Structures - Middle Exam, Nov. 19, 2012

1. What is the meaning of each of the following terms in C++ language? (9%)

   (a) protected

   (b) destructor

   (c) template

2. Interpret the bit string 10010110 as (a) a binary positive integer, (b) a binary integer in 2's complement, (c) a binary-coded decimal integer. What are the values of these interpretations? (6%)

3. Suppose that a matrix $m[\,][\,]$ is stored in a linear array $a[\,]$ with the following sequence:

0	1	8	9	$\cdots$
3	2	7	10	
4	5	6	11	
15	14	13	12	
16	$\cdots$			

   Please give the mapping function from $m[i][j]$ to $a[k]$. Note that the upper left corner of m is the first element $m[0][0]$, the first element of a is $a[0]$, $m[0][1] = 1$ and $m[0][2] = 8$. (8%)

4. Transform the *prefix* expression $+-A*+BC-DE/-FG*HI$ to *infix* and *postfix* expressions. Draw its expression tree. (9%)

5. (a) With a linear scan scheme, how do you check whether a postfix expression is valid or not? (5%)

   (b) Give the algorithm for transforming an infix expression to a postfix expression with a stack. Note that you need not write a C program.(9%)

6. What are printed by the following C program? (6%)

```
int a[ ]={7,8,9};
int *p;
p=a;
*(p++)=5;
printf("%d %d %d %d %d\n",a[0],a[1],a[2],*p,*(p+1),*(p++));
```

7. The *Super Fibonacci sequence* is defined recursively as follows:

$$f(n) = n \text{ if } n = 0, 1, 2$$
$$f(n) = f(n-1) + f(n-2) + f(n-3) \text{ if } n \geq 3.$$

   Assume that $f(0)$, $f(1)$ and $f(2)$ are given.

(a) What is the value of $f(6)$? (3%)

(b) Suppose we use an iterative method to compute $f(n)$. How many additions are required? (3%)

(c) Suppose our program is written recursively for computing $f(n)$. How many additions are required? Please derive a general pattern. (6%)

8. Write a *recursive* C function to perform *binary search* on a sorted array. (12%)

```
#define N 100
int x; /* the element that we want to search */
int a[N]; /* the array that we want to perform binary search */
int binary(···) /* binary search function. Return -1 if x is not found. */
```

9. Write a C function to insert a new element (integer) with value *x* into a sorted list, which is stored in a linearly linked list with implementation of an array. Note that the smallest element is stored in the front node of the linked list. It is assumed that the list before the insertion contains at least one element. (12%)

```
struct nodetype {
    int info;
    int next;
}
struct nodetype node[100];
void place(int *list, int x)
```

You can call the following two functions directly. In other words, you need not write the programs of these two functions.
(1) *void push(list, x)*: insert a node with value *x* into the front of the list.
(2) *void insafter(q, x)*: insert a node with value *x* after node *q* of the list.

10. Write a C function to delete the *n*th element from a circular doubly linked list, which is implemented by dynamic variables. It is assumed that the number of elements in the list is greater than or equal to *n*. If $n = 1$, then the first node of the list will be deleted. If $n = 1$ and the list has only one node, then the list will become empty. If the list has *n* nodes, then the tail will be deleted. You can use the function $freenode(p)$ to free a node *p*. (12%)

```
struct nodetype {
    int info;
    struct nodetype *left, *right;
}
typedef struct nodetype *NODEPTR;
void delete(NODEPTR *list, int n)
/* The external pointer points to the tail, not the head, of the list. */
```