Dept. of Computer Science and Engineering, undergraduate
National Sun Yat-sen University
Data Structures - Middle Exam, Nov. 30, 2008

1. What is the meaning of each of the following terms in C++ language? (12%)

   (a) overloading

   (b) protected

   (c) constructor

   (d) public

2. We declare an array as *int* $k[m][n]$. Assume that each element of array *k* occupies 4 units of storage. Suppose the addresses of $k[3][4]$ and $k[2][7]$ are 404 and 544, respectively. Note that the first element of *k* is $k[0][0]$.

   (a) Is array *k* in *row-major* or *column-major*? Why? (5%)

   (b) What is the address of $k[5][6]$? What is the address of $k[0][0]$? (6%)

   (c) What are the values of *m* and *n*? If you can not determine the values, please explain your reason. (5%)

3. Transform the *prefix* expression $*+A/BC+D*/-EFGH$ to *infix* and *postfix* expressions. Draw its expression tree. (10%)

4. (a) Give the recursive definition of *prefix* expressions. (5%)

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

5. What are the advantage and disadvantage of a linked list implemented by an array and dynamic variables, respectively? (6%)

6. There is a C function in the following:

```
int f(int a, int m)
{
    int t, ans;

    ans=1;
    t=a;
    while(m>0) {
        if (m % 2 ==1)      /* e.g. 7%2=1, 6%2=0 */
            ans = ans * t;
```

```
        t = t * t;
        m = (int) (m / 2);    /* e.g. 7/2=3, 6/2=3 */
    } /* end of while() */
    return(ans);
}
```

(a) What is the answer of $f(2,5)$? (4%)

(b) Use arithmetic expressions or simple sentences to describe what the function of f is. (6%)

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

```
#define N 100
int x; /* the element we want to search */
int a[N]; /* the array we want to perform binary search */
int binary(···) /* binary search function */
```

8. Write a C function to delete the *n*th element from a linearly linked list, which is implemented by dynamic variables. You can assume that the length of the list is greater than *n*. Note that the first element is deleted when $n = 1$. (12%)

```
struct nodetype {
    int info;
    struct nodetype *next;
}
typedef struct nodetype *NODEPTR;
void delete(NODEPTR *list, int n)
```

9. Write a C function to combine two ordered lists into a single ordered list, where these lists are represented with circular doubly linked lists and implemented by an array. Note that each list may be empty. (12%)

```
struct nodetype {
    int info;
    int left, right;
}
struct nodetype node[100];
void combine(int *lista, int *listb)
/* after the combination, the resulting sorted list is pointed by lista */
```