Dept. of Computer Science and Engineering, undergraduate
National Sun Yat-sen University
Data Structures - Middle Exam, Nov. 16, 2009

1. A *band matrix* a is an n-by-n array in which $a[i][j] == 0$, if $|i-j| \geq k$, where k is a positive integer constant and $1 \leq k \leq n-1$.

   (a) What is the maximum number of nonzero elements in such a matrix? Write your answer in terms of n and k. (5%)

   (b) Suppose that these elements (nonzero elements) of the matrix is stored in a one-dimensional array b with the *row-major* way sequentially. Develop an algorithm for mapping $a[i][j]$ to array b, where $|i-j| < k = 2$. Note that the first element of array a is $a[0][0]$, which is stored in $b[0]$. That is $a[0][0] = b[0]$, $a[0][1] = b[1]$, and $\cdots$. (10%)

2. Transform the *postfilx* expression $AB-C+DEF+-GH+*/$ to *infix* and *prefix* expressions. Draw its expression tree. (10%)

3. (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%)

4. (a) In a circular queue with array implementation, where do the pointers *front* and *rear* point to? (4%)

   (b) In a circular queue with array implementation, what are the conditions of an empty queue and a full queue? (6%)

5. In the Towers of Hanoi problem, there are three pegs and n disks with different radius. How many disk movements are required to move the n disks from the source peg to the destination peg? Please derive the formular and give the answer. (10%)

6. What are printed by the following C program? (9%)

```
int *p, *q, *r;
p=(int *) malloc(sizeof(int));
q=(int *) malloc(sizeof(int));
*p=6;   r=q;   *q=8;   q=p;
printf(”%d %d %d\n”,*p,*q,*r);
p=(int *) malloc(sizeof(int));
r=(int *) malloc(sizeof(int));
*p=7;   q=r;   *r=5;   r=p;   *r=9;
printf(”%d %d %d\n”,*p,*q,*r);
```

7. 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 */
```

8. Write a C function to insert a new element (integer) into a sorted list, which is stored in a linearly linked list with implementation of dynamic variables. Note that the smallest element is stored in the front node of the linked list. You can use the following two functions directly, $push(list,x)$: insert a node with information x into the front of the list; $insafter(q,x)$: insert a node with information x after node q of the list. In other words, you do not need to write the programs of these two functions. (12%)

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

9. Write a C function to *concatenate* two *circular linearly linked lists*, which are implemented by arrays. Note that each list may be empty. (12%)

```
struct nodetype {
    int info;
    int next;
}
struct nodetype node[100];
void concat(int *lista, int *listb)
/* lista and listb are the pointers of the two lists */
```