Dept. of Computer Science and Engineering, undergraduate
National Sun Yat-sen University
Data Structures - Middle Exam, Nov. 22, 2010

1. 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 $k[4][3]$ and $k[2][6]$ are 274 and 374, 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 $k[0][0]$? What is the address of $k[3][4]$? (5%)

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

2. In a base *b* number system, the digit set is $\{0,1,2,\cdots,b-1\}$. For example, suppose $b=8$, then $265 = 2*8^2+6*8^1+5*8^0 = 181$ (decimal, $b=10$). In a *signed-digit* system $SD(b,a)$, the base is still *b*, however the negative digits are also included in the digit set $\{-a,-(a-1),\cdots,-1,0,1,\cdots,(a-1),a\}$. For example, the digit set of $SD(10,6)$ is $\{-6,-5,-4,-3,-2,-1,\ 0,1,2,3,4,5,6\}$. And, decimal 98 in $SD(10,6)$ is written as $<1,0,-2>$, since $98 = 1*10^2+0*10^1-2*10^0$ and each digit of $<1,0,-2>$ is in the digit set.

   (a) Convert 96 with $b=10$ to the number with $b=5$. (5%)

   (b) Convert 181 with $b=10$ to the number in $SD(10,6)$. (5%)

   (c) Convert 181 with $b=10$ to the number in $SD(8,4)$. (5%)

3. Transform the *prefix* expression $*-A+**BCD/+EF*GHI$ to *infix* and *postfix* expressions. Draw its expression tree. (9%)

4. Explain the algorithm for converting an *infix* expression into a *postfix* expression with the *stack* structure. Note that you need not write a C program. (10%)

5. What will be printed after the following C program segment is executed? (5%)

```
int a[ ]={6,7,8};
int *p, *q;
p=a;
*(p++)=3;
q=(int *) malloc(sizeof(int));
q=p;
*(++q)=4;
printf("%d %d %d %d \n",*p,*q,a[0],a[1],a[2]);
```

6. Suppose that we have a permutation $p_1 p_2 \cdots p_n$ of $123 \cdots n$. g_i, $2 \leq i \leq n$, is a *pancake* operation which reverses the elements between positions 1 and i. That is, after g_i is performed, the permutation will become $p_i p_{i-1} \cdots p_1 p_{i+1} p_{i+2} \cdots p_n$. For example, after pancake operation g_4 is performed on permutation 634215, the result is 243615.

   (a) What pancake operations are required to transform 31452 into 12345? Give the operation sequence you use. (5%)

   (b) Design a method to transform a permutation $p_1 p_2 \cdots p_n$ into $123 \cdots n$ by using pancake operations. (7%)

7. The C expression *m* % *n* yields the remainder of *m* upon division by *n*. Write a *recursive* C function to find the greatest common divisor (GCD) of two integers. (10%)

8. Write a C function to delete all nodes of value *x* in a *linearly linked list* with implementation of dynamic variables. Note that the number of nodes with value *x* may be zero or greater than one. (12%)

```
struct nodetype {
    int info;
    struct nodetype *next;
}
typedef struct nodetype *NODEPTR;
void delete(NODEPTR *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) *int delafter(NODEPTR q)*: delete the node after node *q* and return the information of the deleted node.
   (2) *int pop(NODEPTR *list)*: remove the front node (head node) of the list.

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 */
/* after concatenation, lista points to the resulting list */
```