Dept. of Computer Science and Engineering, undergraduate
National Sun Yat-sen University
Data Structures - Middle Exam, Nov. 18, 2013

1. Assume that each *int* element of an array a occupies 4 units of storage and each *float* element of an array a occupies 8 units of storage. Suppose that the first element of a is $a[0][0]$ or $a[0][0][0]$ and its address is 500. Please give the address of the indicated element in each of the following cases. (12%)

   (a) int a[8][15]; row-major order; find element $a[3][5]$.

   (b) float a[8][15]; column-major order; find element $a[3][5]$.

   (c) float a[6][4][5]; row-major order; find element $a[4][3][2]$.

   (d) int a[6][4][5]; column-major order; find element $a[4][3][2]$.

   answer:
   (a) 700 (b) 844 (c) 1276 (d) 780

2. What are printed by each of the following C programs? (20%)

   (a)
   ```
   int c=11;
   printf("%d \n",((c >> 3) << 2) +((c <<2) >> 3));
   ```
   answer: 9

   (b)
   ```
   long long int s=1; int i;
   for (i=1; i <= 30; i++) s*=5;
   printf("%d \n",s%7);
   ```
   answer: 1

   (c)
   ```
   union {
   char m;
   unsigned char n; }u;
   u.n=195; printf("%d \n",u.m);
   ```
   answer: -61

   (d)
   ```
   int a[ ]={4,5,6,7,8}; int *p;
   p=a+1; *(++p)=9;
   printf("%d %d %d ",a[2],a[3],*(p+1));
   printf("%d \n",*(p++));
   ```
   answer: 9 7 7 9

   (e)
   ```
   void fun(int a[ ], int b[ , int c[ ], int d[ ])
   { printf("%d %d %d %d \n", a[3],b[3],c[3],d[3]); }
   void main( )
   { int e[ ]={10,11,12,13,14,15,16,17,18,19,20};
   fun(e,e+2,&e[3],&e[3]+2); }
   ```
   answer: 13 15 16 18

3. (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%)

4. Explain the algorithm for transforming an infix expression to a postfix expression by using a stack with the example $((A-(B+C))*D)/E$. You have to give the explanation and show the content of the stack whenever its content is changed. Note that you need not write a C program.(10%)

5. The *Fibonacci Plus sequence* is defined recursively as follows:

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

   Assume that $f(0)$ and $f(1)$ 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%)

   answer:
   (a) f(2)=2, f(3)=4, f(4)=7, f(5)=12, f(6)=20
   (b) 2(n-1)
   (c) g(n)=0, n=0,1
   g(n)=g(n-1)+g(n-2)+2, n¿=2

6. 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. */
```

7. 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.

8. Write a C function to reverse a circular doubly linked list, so that the last element becomes the first, and so on. The list is implemented by dynamic variables. It is assumed that the number of elements in the list is greater than 2. (12%)

```
struct nodetype {
   int info;
   struct nodetype *left,*right;
}
void reverse(....)
```