科目：資料結構(3002)　　校系所組：中央大學電機工程學系（電子組）
交通大學電子研究所（乙A組）

1. [7 points] The following C++ function implements a selection sort algorithm, which sorts *n* integers, from a[0] to a[n-1], in an increasing order.

```
void selectionSort(int *a, const int n )
{
    for(int i=0; i<n; i++){
        int max = I
        for(int j = i+1; j<n; j++){
            if(a[j]>a[max])
                max = j;
        }
        Int temp = a[n-1-i];
        a[n-1-i] = a[max];
        a[max] = temp;
    }
}
```

(a) [2 pts] Assume that n=10. Show the input data from a[0] to a[9] which can result in the worst-case runtime for the above function.

(b) [5 pts] What is the worst-case time complexity of the above selection-sort algorithm?

2. [12 points] Indicate which line of the following C++ codes contains an error and why. Report "No Error" if no error exists. Assuming that <iostream> is included and the namespace std is used.

```
Line 1     class BaseClass {
Line 2         public:
Line 3             BaseClass(int a, int b, int c):
Line 4             aaa(a),
Line 5             bbb(b),
Line 6             ccc(c)  {};
Line 7             int function1() {return aaa;}
Line 8             int aaa = 0;
Line 9         private:
Line 10            int bbb;
Line 11            int function2() {return aaa+bbb;}
Line 12        protected:
Line 13            int ccc;
Line 14            int function3() {return bbb+ccc;}
Line 15    };
Line 16    class DerivedClass: public BaseClass {
Line 17        public:
Line 18        DerivedClass(int a, int b, int c, int d){
Line 19            BaseClass(a,b,c);
Line 20            ddd=d;
Line 21        }
```

注意：背面有試題

```
Line 22         void setABCD(int a, int b, int c, int d){
Line 23             aaa=a;
Line 24             bbb=b;
Line 25             ccc=c;
Line 26             ddd=d;
Line 27         }
Line 28     private:
Line 29         int ddd;
Line 30         void print(){
Line 31             cout << function1() << endl;
Line 32             cout << function2() << endl;
Line 33             cout << function3() << endl;
Line 34         }
Line 35     };
Line 36     int main()
Line 37     {
Line 38         const BaseClass a(0,2,4);
Line 39         BaseClass b(1,3,5);
Line 40         BaseClass c;
Line 41         DerivedClass d(2,4,6,8);
Line 42         cout << a.function1() << endl;
Line 43         d.setABCD(7,8,9,10);
Line 44     }
```

3. [12 points] Consider the prefix, infix, and postfix expressions.
Following are the examples of representing a+b in prefix, infix, and postfix expressions.

- Prefix: + a b
- Infix: a + b
- Postfix: a b +

(a) [6 pts] The following *infixToPostfix* function can convert an infix expression (including parentheses) into a postfix expression.

```
bool opGreaterEqual(char a, char b){
    return (a=='*' || a=='/') >= (b=='*' || b=='/');
}
string infixToPostfix(string s){
    char temp;
    string ans="";
    stack<char> stack;
    for(int i=0; i<s.length(); i++)  {
        if (s[i]=='+' || s[i]=='-' || s[i]=='*' || s[i]=='/'){
            while( !stack.empty() && stack.top()!='(' ){
                if ( opGreaterEqual(stack.top(), s[i]) ){
                    temp = stack.top();
                    stack.pop();
                    ans+=temp;
                }
                else break;
            }
            stack.push(s[i]);
        }
        else if ( s[i]=='(' ) stack.push('(');
        else if ( s[i]==')' ){
            while ( !stack.empty() && stack.top()!='(' ){
               temp = stack.top();
                stack.pop();
                ans+=temp;
            }
            if ( !stack.empty() ) stack.pop();
        }
        else if ( s[i]>='A' && s[i]<='Z' )   {
            ans+=s[i];
        }
    }
    while ( !stack.empty() ){
       temp = stack.top();
        stack.pop();
        ans+=temp;
    }
    return ans;
}
```

Assume that the input parameter s is set to "A * (B + C) – D / E" for the above function. List the value of s[i], stack, and ans at the end of each iteration of the *for* loop in the format of the following table. Also, list the value that the function returns.

i	s[i]	stack (top -> bottom)	ans
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Returned ans =

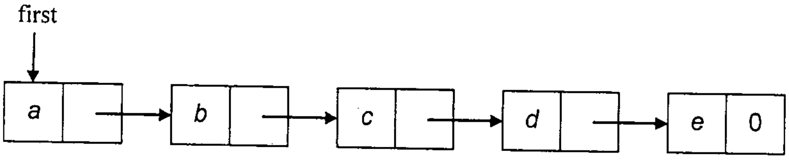
(b) [6 pts] Modify the above function code to convert an infix expression (including parentheses) into a prefix expression. You need to highlight the code you modified in your answer.

4. [12 points] A template class is defined as below and the current condition in memory is shown as the following figure. Please write down a pseudo code to complete the following process:

```
template <class T>
class ChainNode
{
  private:
  T data;
  ChainNode<T> *link;
  ChainNode() {}     // Constructor 1
  ChainNode(const T& data)     // Constructor 2
  {this -> data = data;}
};
```

(a) [6 pts] Insert a node "*f*" between the third "*c*" and fourth "*d*" nodes.

(b) [6 pts] Delete the first node (please remember to clear unused memory).



5. [15 points] Trees

(a) [5 pts] What is the maximum number of nodes in a k-ary tree of height h? Prove your answer.

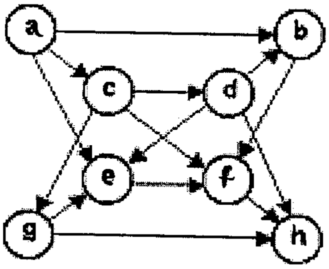
(b) [5 pts] For any nonempty binary tree, T, if n_0 and n_2 are the number of leaf nodes and the number of nodes of degree 2, respectively. Prove that $n_0 = n_2 + 1$.

(c) [5 pts] Prove that every binary tree is uniquely defined by its preorder and inorder sequences.

6. [10 points]

(a) [5 pts] If there are no articulation points in a graph *G*, what kind of property does *G* have?

(b) [5 pts] List the order in which the nodes of the following directed graph are visited by a topological order traversal that starts from vertex *a*.



注意：背面有試題

7. [12 points]
   (a) [6 pts] Given a connected, undirected graph, develop an algorithm (in pseudo code) to find a spanning tree of the minimum height. Argue that your algorithm is correct.
   (b) [6 pts] Let G be a directed weighted graph, and let s and t be two vertices in G. Define the bottleneck $b(p)$ of a path p to be the minimum-weight edge on the path; that is, $b(p)=min_{e\in p}w(e)$. Describe how you could modify Dijkstra's algorithm to find a path from s to t with the maximum bottleneck.

8. [12 points] For each of the following statements to indicate whether the statement is true or false, respectively. Justify your answer briefly. No points will be given even for a correct answer if no justification is presented.
   (a) [3 pts] Insertion sort runs longer than Merge sort on all inputs.
   (b) [3 pts] The worst-case runtime and expected runtime are equal to within constant factors for any randomized algorithm.
   (c) [3 pts] The fastest way to find the median of an unsorted array is to first sort the array, and then output its middle element.
   (d) [3 pts] Suppose that an array contains n numbers, each of which is -1, 0, or 1. Then, the array can be sorted in O(n) time in the worst case.

9. [8 points] Let a hash table and a hashing function be defined as follows:
   table: array[0..7] of hashnode
   The hashing function is selected as:
   h(x) = mid_squaring(x)
   where x is calculated from x (an alphanumeric string) by summing the numbers which represent the alphanumeric characters. The A..Z alphabets will be represented as numbers 1..26 and 0..9 be as numbers 0..9.
   For example,
   AZ6 = 1 + 26 + 6 = 33
   Mid-squaring is calculated by
   (1) $y = x^2 = (33)^2 = 1089$
   (2) Take left 3-bit starting from the second right most bit of y.
   $Y = 1089 = (1000100\boxed{000}1)_2$

   The following operations are performed starting with an empty table.
   insert 1B4
   insert F7W
   insert Y48
   insert 8B5
   insert MSN
   insert DF3
   insert ABC

   With linear probing as rehash technique, draw the final hash table.