國立臺灣師範大學113學年度碩士班招生考試試題

科目：**軟體基礎**　　適用系所：**資訊工程學系**

注意：1.本試題共 7 頁，請依序在答案卷上作答，並標明題號，不必抄題。2.答案必須寫在指定作答區內，否則依規定扣分。

1. (5 points) Please write down the output of the following code.

```
1  #include <stdio.h>
2
3  void f1( int a )
4  {
5          if( a > 0 ) f1( a / 2 );
6          printf( "%d", a % 2);
7  }
8
9  int main()
10 {
11         f1( 12345 );
12         return 0;
13 }
```

2. (5 points) Please write down the output of the following code.

```
1  #include <stdio.h>
2
3  enum state
4  {
5          STATE_00 = 0,
6          STATE_01 ,
7          STATE_02 ,
8          STATE_03
9  };
10
11 int f1( int s ) { return s == 0 ? STATE_01:STATE_02; }
12 int f2( int s ) { return s == 0 ? STATE_03:STATE_00; }
13 int f3( int s ) { return s == 0 ? STATE_00:STATE_03; }
14 int f4( int s ) { return STATE_03; }
15
16 int main()
17 {
18         int (*func [4])(int);
19         func [0] = f1;
20         func [1] = f2;
21         func [2] = f3;
22         func [3] = f4;
23
24         int input [10] = { 1, 0, 0, 1, 1, 0, 0, 1, 1, 0 };
25         enum state s = STATE_00;
26
27         for( int i = 0 ; i < 10 ; i++ ) { s = func[s]( input[i]); }
28         printf( "%d", s );
29         return 0;
30 }
```

3. (5 points) Please describe the following function f3.

```
1 void f3( char *s )
2 {
3         while( *s != 0 )
4         {
5                 *s = *s >= 'A' && *s <= 'Z' ? *s += 'a' - 'A' : *s;
6                 s++;
7         }
8         return;
9 }
```

4. (5 points) Please see the following two codes.

```
1 #include <stdio.h>
2
3 int main()
4 {
5         char str[] = "Hello Kitty";
6         str [6] = 'W';
7         printf( "%s\n", str );
8         return 0;
9 }
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5         char *str = "Hello Kitty";
6         str [6] = 'W';
7         printf( "%s\n", str );
8         return 0;
9 }
```

The first program will output "Hello Witty" while the second program will output "Segmentation Fault". Please describe the reason of this failure.

5. (5 points) gets() is a standard C function, which get a string from standard input. The synopsis is as follows.

```
1 #include <stdio.h>
2
3 char *gets(char *s);
```

gets() reads a line from stdin into the buffer pointed to by s until either a terminating newline or EOF, which it replaces with a null byte. gets() returns s on success, and NULL on error or when end of file occurs while no characters have been read.

Actually, this function is deprecated and should not be used. Please explain the reason why this function should never be used.

6. (5 points) Please see the following C code and describe the purpose of macro TEST.

```
#define OFF(TYPE , MEMBER) (( size_t)&(( TYPE *)0)->MEMBER)

#define TEST(ptr , type , member) ({    \
    void *__mptr = (void *)(ptr);    \
    (( type *)(__mptr - offsetof(type , member))); })
```

7. (10 points) Do you know how to play GO (圍棋)? I will teach you some basic knowledge of GO. A liberty (氣) is a vacant point that is immediately adjacent to a stone(棋子) in a cardinal (orthogonal) direction, or connected through a continuous string of same-colored stones to such a point. Fig. 1 is an example.

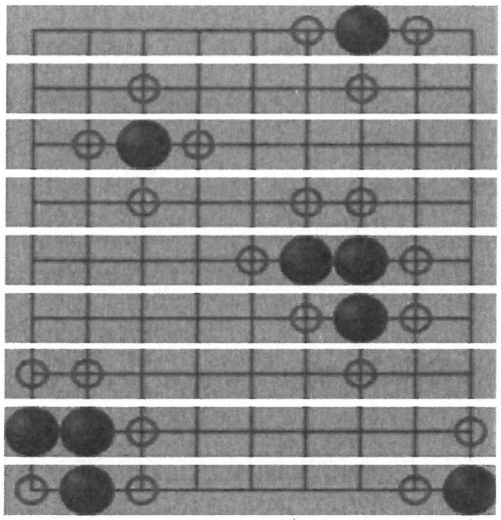


Figure 1: Groups and their liberties (marked by circles)

Please develop a function, where the prototype is as follows, to count the liberty number of a stone or a chain pointed by the user.

```
// Input:
// int board [19][19]: 19*19 board , 0: vacant point , 1: black , 2: white
// int x, int y: the given position.
// For your simplicity , you can assume the given point is the black stone.
int count( int board [19][19] , int x, int y );
```

8. (7 points) Please answer the following questions. For (a) and (b), you should give reasonable explanation for your answers to get the score.

(a) (2 points) What is the worst-case time complexity for **finding the height** of a binary tree with *n* nodes?

(b) (2 points) In a binary tree, what is the **maximum number of leaf nodes** possible in a tree with *n* nodes?

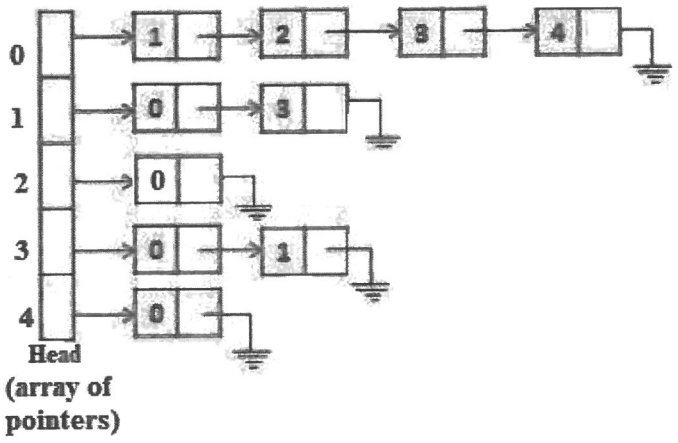
(c) (3 points) Given the type definition of a node in a tree:

```
typedef struct TreeNode *tree_Pointer;
typedef struct TreeNode{
   tree_Pointer leftChild, rightChild;
   char data;
   };
```

The following C program codes will test the equality of two binary trees, where *first* and *second* are pointers of the root nodes. If the two trees are equal (the tree structure and data stored in notes in both trees are all the same), the function will return 1. Otherwise, 0 will be returned. Please fill in the blanks.

```
int equal (tree_Pointer first, tree_Pointer second)
{
   return (( !first&& !second) || (_____<1>_____&&
           (_____<2>_____) &&
           equal(first->leftChild == second->leftChild) &&
           equal(_____<3>_____)))
```

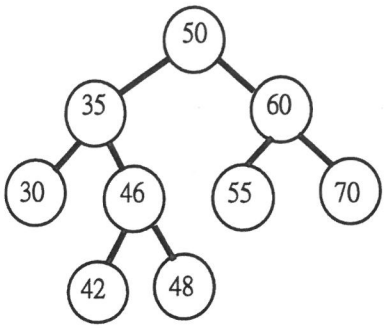
9. (7 points) Given an adjacent list of a graph of a graph G1 as shown below:



(a) (4 points) Please draw the DFS and BFS spanning trees starting from vertex 2.

(b) (3 points) Please draw the bi-connected components.

10.(5 points) Please show the resultant AVL tree after inserting 40, 10, 20, and 32 to the AVL tree below. Please show the 4 AVL trees after each insertion.



11.(5 points) Please fill in the blanks to implement the following C program codes by adding the ***item*** into a queue and deleting from a queue by a singly linked list whose two ends are pointed by ***front*** and ***rear***, respectively. If the queue is empty, ***front*** = null and ***rear*** = null.

```
void addq (item) {
    queuePointer temp;
    MALLOC (temp,
sizeof(*temp));
    temp->data = item;
    temp->next = _____<1>_____;
    if (front)
        _____<2>_____;
    else
        _____<3>_____;
    rear = temp;
}
```

```
element deleteq() {
    queuePointer temp = front;
    element item;
    if (!temp)
        return queueEmpty();
    item = temp->data;
    front = _____<4>_____;
    free(_____<5>_____);
    return item;
}
```

12.(6 points) The following C++ codes implement a function according to the Dijkstra algorithm to get the cost of the shortest path from vertex *v* to all the other vertices in an array distance[]. The costs of the directed edges are stored in the adjacency matrix cost[][MAX_VERTICES] and the given costs are non-negative integers. *n* denotes the number of nodes in the graph. Please complete the codes in the blanks.

```
int choose (int distance[], int n, short in found[])
{    /* find the smallest distance not yet checked */
    int i, min, minpos;
    min = INT_MAX;
    minpos = -1;
    for (i=0; i<n; i++)
        if (_____<1>_____< min && !found[i]) {
            min = distance[i];
            minpos = _____<2>_____;
        }
    return minpos;
}
```

```
void shortest_path (int v, int cost[][MAX_VERTICES],
                    int distance[], int n, short int found[])
{
    int i, u, w;
    for (i=0; i<n; i++) {
        found[i] = FALSE;
        distance[i] = cost[v][i];
    }
    found[v] = TRUE;
    distance[v] = 0;
    for(i=0; i<n-2; i++) {
        u =choose(distance, n, found);
        found[u] = TRUE;
        for(w=0; w<n; w++)
            if (_____<3>_____)     /* compute the distance by u
                if (distance[u]+cost[u][w]<_____<4>_____)
                    distance[w]= distance[u]+cost[u][w];
    }
```

13.(30 points) Please answer the following questions regarding bipartite graph and maximum matching.

(a) (5 points) You are given an adjacency list representation of an undirected graph as follows.

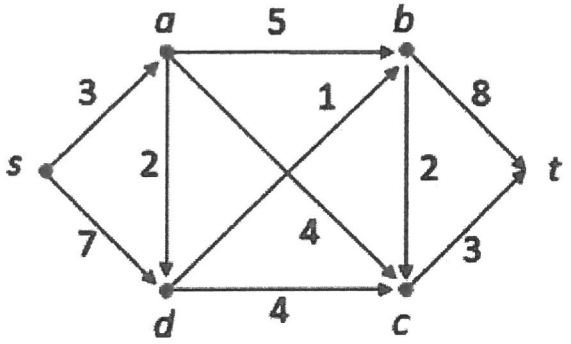
```
G = {
        a -> b, c
        b -> a, d, e, f
        c -> a, d, e
        d -> b, c
        e -> b, c
        f -> b
}
```

Please traverse this graph by using depth-first search (DFS). When you traverse the neighbors, process them in alphabetical order. Please start from vertex `a`.

(b) (5 points) A graph is ***bipartite*** if all of its vertices can be partitioned into two disjoint subsets X and Y so that every edge connects a vertex in X with a vertex in Y. Is the graph given in (a) bipartite? If your answer is yes, please show the partition of vertices.

(c) (5 points) Please provide an algorithm to check if a graph is bipartite using DFS.

(d) (10 points) Given a network with a source s and a sink t, the Ford-Fulkerson algorithm for solving the maximum-flow problem iteratively finds an augmenting path in a residual network. Now we have an input network shown below. Please find the maximum flow and the minimum cut of the network. For the minimum cut, please show the partition of vertices of the cut.



(e) (5 points) A matching in a bipartite graph is a set of the edges chosen in such a way that no two edges share an endpoint. A maximum matching is a matching of maximum size (maximum number of edges). Please describe how the Ford Fulkerson method can be used to solve the maximum bipartite matching problem.