國立中正大學
109學年度碩士班招生考試
試 題

［第2節］

科目名稱	軟體設計
系所組別	資訊工程學系－甲組

—作答注意事項—

※作答前請先核對「試題」、「試卷」與「准考證」之系所組別、科目名稱是否相符。

1. 預備鈴響時即可入場，但至考試開始鈴響前，不得翻閱試題，並不得書寫、畫記、作答。

2. 考試開始鈴響時，即可開始作答；考試結束鈴響畢，應即停止作答。

3.入場後於考試開始40分鐘內不得離場。

4.全部答題均須在試卷（答案卷）作答區內完成。

5.試卷作答限用藍色或黑色筆（含鉛筆）書寫。

6. 試題須隨試卷繳還。

(5%) Questions 1 to 3. Circle **T** for true or **F** for false.

1. (1%) (   ) A class can have more than one derived class.

2. (2%) (   ) A class can have more than one destructor.

3. (2%) (   ) If a C++ class definition does not include a constructor, the compiler will automatically provide a default constructor.

4. (20%) Below is a **bubble sort** program that sorts an integer array containing 10 elements. The algorithm sorts elements in **ascending order**, i.e., the greatest element in the list will be carried to the end of the list. However, the first version of the program contains a lot of bugs. Please indicate and correct all bugs so that the program creates the desired output.

   Try to use as **few changes** as possible to make the program compile and run correctly. You may also insert and delete lines if you like to. Do not rewrite entire lines of code but try to keep the changes as **small** as possible.

```
Line 1:     #include <iostream>
Line 2:     #include <vector>
Line 3:     using namespace std;
Line 4:
Line 5:     void SwapIntegers (int a, int b) {
Line 6:         int temp = a;
Line 7:         a = b;
Line 8:         temp = b; }
Line 9:
Line 10:    void BubbleSort (vector &intVector) {
Line 11:        for (int i = intVector.size() - 1; i > 0; i++)
Line 12:            for (j = 0; j < i; j++)
Line 13:                if (intVector[j] < intVector[j + 1])
Line 14:                    SwapIntegers (intVector[j], intVector[j + 1]); }
Line 15:
Line 16:    int main () {
Line 17:        int intArray[] = {34, 1432, 1, -54, 16, 22, 13245, 512, -3000, 0};
Line 18:        vector<int> intVector(intArray, intArray + 9);
Line 19:
Line 20:        BubbleSort (&intVector);
Line 21:
Line 22:        for (int i = 0; i <= intVector.size(); i++)
Line 23:            cout >> intVector[i] >> endl;
Line 24:
Line 25:            return 0; }
```

5. (10%) Please write a function to perform linked list insertion. Assume that each node has a name field with at most 32 bytes, and a link field. You have to declare the data structure for the node. The inserted node will become the first node of the linked list. The return value of the function is a pointer to the head of the linked list.

6. (5%) Please finish the following code

```
#include <stdio.h>

void swap(        ,        )
{
}

void main()
{
    int x, y;
    x = 10;
    y = 20;

    swap(      ,      );
    printf("x=%d, y=%d\n", x, y);
}
```

After calling swap function, the content of x and y is expected to be 20, 10. That is, the output of the program is

```
x=20, y=10
```

7. (10%) Please write a program to read data from stdin and output the longest line in the input data. You have to use fgets() function to perform the input operation. If there are more than one longest lines, output the first one.

For example, if the input data is as follows

```
apple
banana
abc
string
```

The output of the program will be

banana

8. (4%) Please transform this expression from the infix to the postfix.
   (a) (2%) a*(b+c/d)-e
   (b) (2%) (a+c)*b+d*e

9. (4%) Find the numbers of different binary trees with 2, 3, 4, and 5 nodes, respectively.

10. (8%) You are given the following eight numbers: 50, 10, 80, 90, 70, 60, 65, 62. Please draw the final result after the numbers are inserted to a AVL tree and a Red-Black Tree, respectively.

11. (4%) Please write down the final result of failure array obtained by the following Program (i.e., fail function) for each of the following patterns (i.e., pat).
   (a) a b a a b a a b
   (b) a b c a b a b c a b c

```
void fail (char *pat, int *failure)
{
  int i, j, n = strlen(pat);
  failure[0] = -1;
  for(j = 1; j<n; j++)
  {
    i = failure[j-1];
    while (pat[j] != pat[i+1] && (i >= 0))
      i = failure[i];
    failure[j] = (pat[j] == pat[i+1]) ? (i+1) : -1;
  }
}
```

12. (5%) You are given the following ten numbers: 55, 45, 25, 35, 85, 95, 65, 75, 105, 15. Please sort these numbers using quick sort, where the first element of a sublist is always picked as the pivot. Note that you have to write the number sequence for each pass.

13. (18%) True or false. If the statement is false, correct the wrong part. Simply negate the statement is not accepted. (3% each)

(1) Given an array A[1 . . n] of integers, the running time of Counting Sort is polynomial in the input

size n.

(2) Given an array A[1 . . n] of integers, the running time of Heap Sort is polynomial in the input size n.

(3) Depth-first search will take Θ(V^2) time on a graph G = (V, E) represented as an adjacency matrix.

(4) For every dynamic program, we can assign weights to edges in the directed acyclic graph of dependences among subproblems, such that finding a shortest path in this DAG is equivalent to solving the dynamic program.

(5) If a problem X can be reduced to a known NP-hard problem, then X must be NP-hard.

(6) Using DFS to search augmenting paths in Ford-Fulkerson algorithm could reduce the time complexity

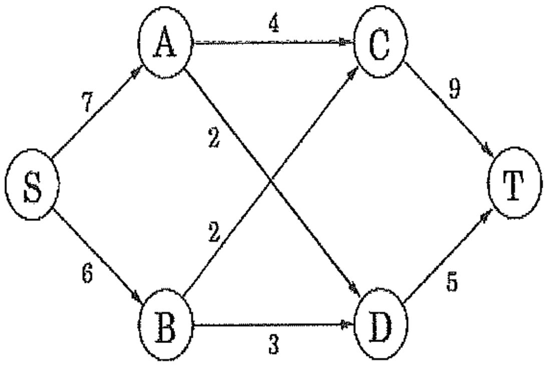
14. (7%)

Given a network at the right side figure (the numbers are edge capacities),

(a) (3%) Find its maximum flow and a minimum cut.

(b) (2%) Draw its residual graph. In your answer mark the vertices reachable from S and the vertices from which T is reachable.

(c) (2%) An edge of a network is called a bottleneck edge if increasing its capacity results in an increase in the maximum flow. List all bottleneck edges in the network.


A
4
C
7
9
2
S
T
2
6
5
B
3
D