1. (10%) Consider the following C statements.

```
i = 3;
j = 4;
k = 5;
x = ++i + j;
if (j < k || k = j - 1) y = j + k;
z = j + (j += 1);
```

(a) (3%) What is the value of variable x after the execution of these statements?

(b) (3%) What is the value of variable y after the execution of these statements?

(c) (4%) The value of variable z after the execution of these statements may be different for different compilers. Why?

2. (12%) In the table below, the names of the three ways to pass parameters in C++ are written in the first row. In the rest of the table, choose "YES" or "NO" to answer the questions for each parameter-pass method.

| Does this method... | BY VALUE | BY REFERENCE | BY REFERENCE-TO-CONST |
|---|---|---|---|
| ... make a copy of the object passed? | (a) YES or NO? | (e) YES or NO? | (i) YES or NO? |
| ... allow const objects to be passed? | (b) YES or NO? | (f) YES or NO? | (j) YES or NO? |
| ... allow for polymorphism/virtual dispatch? | (c) YES or NO? | (g) YES or NO? | (k) YES or NO? |
| ... allow implicit type conversions to be performed? | (d) YES or NO? | (h) YES or NO? | (l) YES or NO? |

3. (3%) Consider the following code that is supposed to recursively count how many ways a set of golf balls can break into two.

```
public static int split(int num) {
  // split even number of golf balls into two:
  if ( num % 2 == 0 )
    return split(num/2) + split(num/2);
  // else split odd number of golf balls by one:
    return 1 + split(num - 1);
}
```

(a) What is the most significant oversight/error in the above code?

(b) When would you expect to see a problem – When you compile and/or execute? (choose the correct answer)

i. Compilation Problem? YES or NO?

ii. Execution Problem? YES or NO

4. (15%) Give the output of the following C++ code.

```
#include <iostream>
using namespace std;

class A
{
   public:
       void show(int x) { cout << cal(x) << endl; };
       virtual int cal(int x) { return x*5; };
};

class B: public A
{
   public:
       int cal(int x) { return x+10; };
};

class C: public B
{
   public:
       int cal(int x) { return (x>20)?(x-10):(x+10); };
};

void func(A a, int x)
{
   a.show(x);
}

int main()
{
   A objA;
   B objB;
   C objC;

   objA.show(10);
   objB.show(20);
   objC.show(30);

   func(objA, 9.99);
   func(objB, 20.4);

   return 0;
}
```

5. (10%) The C preprocessor is designed to extend the C programming language.
   (a) (3%) Consider the following C preprocessor directive .

```
#define        max(m, n)      m >= n ? m : n
```

   What does this directive mean?

   (b) (3%) Consider the following C statements that use the above C preprocessor directive.

```
y = z = 8;
x = max(y , z) - y - z;
```

   What is the value of variable x after the execution of these statements?

   (c) (4%) From (b), we know there are errors in the above C preprocessor directive. Correct the above C preprocessor directive.

6. (6%) Which are not true for Open Addressing?
   (a) may cause clustering effect
   (b) uses open probing to search
   (c) does not support data deletion
   (d) requires O(log N) search time

7. (8%) Time Complexity:
   (a) What is the worst case time complexity to search in a complete graph with N nodes?
   (b) What is the worst case time complexity to search in a Binary Search Tree with N nodes?
   (c) What is the worst case time complexity to sort N data element using Quicksort?
   (d) What is the worst case time complexity to search an AVL tree with N nodes?

8. (6%) Please describe M-Way Mergesort algorithm for external sorting.

9. (9%) Consider the following algorithm for finding a topological order of a directed acyclic graph G with vertex set V and edge set E, in which Q is a FIFO queue. Fill in blanks (1)~(4). (2 points for each of the first three blanks and 3 points for blank (4).)

```
Algorithm TS
    for each u ∈ V do
        store (___(1)___) in xx[u];
        if xx[u] = 0 then insert u into Q;
    while (Q is not empty)
        u = Extract(Q);
        for each edge (u,v) do
            (___(2)___);
            if ___(3)___ then insert v into Q;
```

The instruction (2) is executed ___(4)___ times in total.

10. (6%) Consider the following algorithm for finding a minimum spanning tree, in which the input is a graph G with edge weight w and a vertex r of G, and Q is a priority queue. Fill in blanks (1), (2) and (3). (2 points for each blank)

```
Algorithm WT(G, w, r)
    for each u ∈ V(G)
        key[u] = ∞;
    key[r] = 0;
    Insert all vertices into Q;
    while (Q is not empty)
        u = ___(1)___;
        for each neighbor v of u and v ∈ Q
            if key[v] < ___(2)___
                key[v] = ___(3)___;
```

11. (5%) What is the asymptotic tight bound (Big O) of the following equation? Justify your answer.

$$T(n)=T(\frac{n}{7})+T(\frac{5n}{7})+n$$

12. (10%) The knapsack problem asks for a maximum packing such that the total value of packed items is maximized, requiring that the total weight should not exceed the knapsack capacity. Consider a knapsack with capacity of four units. And there are four items listed below. You are asked to describe a dynamic programming algorithm for finding the maximum total value of this problem.

| Item | Weight | Value |
|---|---|---|
| 1 | 1 unit | 1 |
| 2 | 1 unit | 2 |
| 3 | 2 units | 4 |
| 4 | 2 units | 3 |