1. (4%) Give the output of the following C++ code.

```
#include <stdio.h>
void main() {
    int i = 1, j = 2;
    int m, n;
    m = i + j * 10;
    n = i++ * 5;
    printf("%d, %d\n", m, n);
}
```

2. (8%) There are 29 days in February in a *leap* year. In a year that is not a leap year, there are 28 days in February. A year *y* is a leap year if *y* is divisible by 400, or *y* is divisible by 4 and is not divisible by 100. Complete the body of the following C++ function `isleap(int y)` that determines if the parameter `y` is a leap year.

```
int isleap(int y) {_________}
```

3. (8%) Complete the three inner for-loops in the following C++ code

```
#include <stdio.h>
void main() {
    int i, j, k;
    for (i = 0; i <= 2; i++) {
        for (j = _____) printf("#");
        for (k = _____) printf("*");
        for (j = _____) printf("#");
        printf("\n");
    }
}
```

to print the following output

```
*****
#***#
##*##
```

4. (5%) Consider the following procedure declaration in C:

```
void p() { int x; q = &x; *q = 1; }
```

where `q` is a global variable declared as pointer to **`int`** and initialized by `q = new int;`. What output can we expect to be produced by the following piece of code:

```
*q = 0; p(); r(); printf ("%d\n", *q);
```

where **`r`** is a procedure which does not mention `q`.

5. (10%) Consider the following program in C++:

```
#include <stdio.h>
int x = 0;
void p(int, int);

void main() {
    int x = 1;
    p (x, x);
}

void p(int y, int z) {
    x = x + 1;
    y = y + 1;
    z = z + 1;
    printf("%d\n", x + y + z);
}
```

   (a) What is the value printed?

   (b) Suppose that we modify the declaration of p so to pass the parameters by reference. Namely, we write the declaration as

   ```
   void p(int &y, int &z) {...}.
   ```

   What is the value printed in this case?

6. (10%)

   (a) Please describe the M-Way external mergesort algorithm.

   (b) What data structure is needed in performing the M-Way merging?

7. (6%)

   (a) Please convert the following infix expression into postfix expression:

   ```
   X - (A + B ) / C
   ```

   (b) What data structure is needed in performing the conversion?

8. (4%) What is the expected time complexity to search in a sorted linked list with *N* data elements?

   (a) O(*N* log *N*)

   (b) O(*N*)

   (c) O(log *N*)

   (d) O(1)

9. (15%) Give the output of the following C++ code.

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

class A
{
  public:
    A() : factor(2) {}
    A(double x) : factor(x) {}
    int mul(double x) { return (x*factor); }
    int div(double x) { return (x/factor); }
  private:
    int factor;
};

class B: public A
{
  public:
    B(double x) : A(2*x), factor(x) {}
    int div(double x) { return (x/factor+0.5); }
    int div(int x) { return (x/factor); }
  private:
    int factor;
};

int main()
{
  A objA;
  B objB(3);

  int first(5);
  double second(13.5);

  cout << objA.mul(first) << "," << objA.div(second) << endl;
  cout << objB.mul(first) << "," << objB.div(second) << ","
       << objB.div(first) << endl;

  return 0;
}
```

10. (8%) Let *G* be a directed acyclic graph (dag) with vertex set *V* and arc set *E*. Suppose that *s* is the only vertex with in-degree zero. Consider the following algorithm for finding the shortest path lengths from *s* to all the other vertices. Here each arc has unit length. Fill in blanks (1)~(4). (2 points for each blank.)

```
Algorithm TTT
    find a (__(1)__) (v1=s,v2,…,vn) of G; // answer what kind of sequence it
is.
    set d[1]= 0 and d[i]= n for any i > 1;
    for i from 1 to n-1 do
        for each j such that there is an arc (vi,vj) do
           if (__(2)__) then (__(3)__)
```

The time complexity is O(__(4)__).

11. (7%) Consider the following recursive algorithm. Fill in the following blanks (3 points for the last blank and 2 points for each of the others).

```
Algorithm WW(i, j)
  if (j - i < 2) return 10;
  else return WW(i+1, j) + WW(i, j-1) + WW(i+1, j-1);
```

To analyze the time complexity, we define `T(n)` to be the time complexity of `WW(i,j)`, where `n = j-i`. Complete the following recurrence relation: `T(n)=` (__(1)__) if `n < 2`; otherwise `T(n)=` (__(2)__). Solving the relation, we have `T(n)=` O(__(3)__).

12. (6%) Compare the Big-O complexity for each pair of expressions below (greater than, less than, or equal to). No explanation required.
    (a) (2%) $\log n$ v.s. $\sqrt{n}$
    (b) (2%) $\log n$ v.s. $\log\log n$
    (c) (2%) $n \log n$ v.s. $\log n!$

13. (4%) Answer Yes or No for the following problems.
    (a) (2%) Is Euler tour problem in NP? No explanation required.
    (b) (2%) If the Travelling Salesman Problem (TSP) is solved in $O(n^{1000})$ time, can we say all other problems in NP can also be solved in polynomial time? No explanation required.

14. (5%) Given two sequences: GTCA and GTACG, find their longest common subsequence (LCS) using dynamic programming.