- **You must submit your work electronically (via moodle) by 11:59 PM of the due date (see above)**.
- **Write your group's own answer and show all your work**.
- **Every assignment must be typed/generated into a pdf file (convert your file into a pdf format before submission).**
- **Weight: 5%**
- **On your assignment, specify the following at the beginning**.
  Assignment number: 3
  Course number: CSC 220 001
  Term: Winter, 2018
  Instructor name: Dr. Manki Min
  Student names: (names of all the members in your group)

Do the following:

1. Consider the following recursive function:

```
int F(int n)
{
    if (n == 0)
        return 0;
    return n + F(n - 1);
}
```

   (a) Find `F(5)`. 15

   (b) Find `F(10)`. 55

   (c) Suppose + is changed to `*` in the inductive step. Now find `F(5)`. 0

   (d) What happens with the function call `F(-1)`?
   it goes on forever or until it reaches the max size of int

2. Determine what is calculated by the following recursive functions (i.e., what do they do in plain english):

   (a)

```
int F(int n)
{
    if (n == 0)
        return 0;
    return n * F(n - 1);
}
```

F takes a number, n, that will multiply itself to n minus one until n equals 0 then it returns 0. The with an positive n number the result will always be 0

   (b)

```
int F(double x, int n)
{
    if (n == 0)
        return 0;
    return n + F(x, n - 1);
}
```

F takes two numbers, x and n, x is not used and is just an extra variable. F takes n and adds it to n minus one until n equals 0, then it returns 0. It results in the sum of n and every positive integer below it.

   (c)

```
int F(int n)
{
    if (n < 2)
        return 0;
    return 1 + F(n / 2);
}
```

F takes a number, n, and when n >= 2 it adds one to the end result and divides n by 2 until n < 2 where it returns 0. The end result is how many times n could be divided by 2 and be greater or equal to 2.

(d)

```
int F(int n)
{
    if (n == 0)
        return 0;
    return F(n / 10) + n % 10;
}
```

F takes a number, n, and as long as n != 0 it adds the digit in the ones place to the end result and divides n by 10. The end result is the sum of each individual digit eg: 123 will return 6

(e)

```
int F(int n)
{
    if (n < 0)
        return F(-n);
    if (n < 10)
        return n;
    return F(n / 10);
}
```

F takes a number, n, and if n is greater 10 it divides it by 10, if n less 10 but greater than or equal to 0 it returns itself, n, if n is less than 0 it returns the positive value of n. The result of this function is if it is greater than 10 it returns the most significant digit, if it is less than 10 it returns itself, if it is less than 0 it returns postive n.

3. Draw each move for hanoi with 5 discs from tower 1 to tower 3. How many moves are required for 5 discs?
   31
4. Suppose that each move of Hanoi takes 1 second. How long will it take to complete Hanoi with 100 discs? Express this in reduced terms (i.e., 86, 401s = 1day, 1sec).
   1,267,650,600,228,229,401,496,703,205,375 seconds or 40,196,936,841,331,475,186.983232032439 millennia
5. Do the following questions for Hanoi.
   (a) What are the *last* three moves for Hanoi with 2 discs?
   (b) What are the *last* three moves for Hanoi with 3 discs?
   (c) What are the *last* three moves for Hanoi with 4 discs?
   (d) What are the *last* three moves for Hanoi with 343 discs?

   The answer for (a) (b) (c) (d)
   Move disc 1 From tower 0 to tower 1
   Move disc 2 From tower 0 to tower 2
   Move disc 1 From tower 1 to tower 2

   (e) What is the complexity of Hanoi (in "big-o" terms)? O(2^n)
6. For this one, you will create **ordered** binary trees from this input:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
9876543210
MFCJABDEHGIKLTPNORSQWUVYXZ
SPHINCTERSAYSWHAT
524137968
MATLSO
FTERFO
EYBLEIF
LYBWIOL
SYGTHO
FPNOEDESO
LLTDREOI
7349AB812
```

Each line contains the input for a single tree. The input for a tree is read from left-to-right and is inserted into the tree, one character at a time. After you have created the tree for an input, print out its *preorder*, *inorder*, *postorder*, and *reverse inorder* traversals. A reverse inorder traversal is simply going to the right, visiting the node, and then going to the left (i.e., RVL). Finally, print the tree out (sideways) as we illustrated in class. **Indent three (3) spaces in between levels.** Make sure to read the input from `stdin` so that I can execute your program, for example, as follows: `java TraversingTheTrees < input`. You are supposed to fill in the codes BinaryTree.java and TraversingTheTrees.java and generate the same output as the provided one using the provided input. The two .java files, input, and output files can be found in moodle.

7. Consider the following list of numbers (input from left-to-right):

   $$13\ 9\ 5\ 7\ 12\ 2\ 3\ 14\ 6\ 1$$

   (a) Create the ordered binary tree, in other words print the binary tree with the above insertion of nodes step-by-step.
   (b) List the preorder traversal of the tree.
   (c) List the inorder traversal of the tree.

8. Consider the following tree traversals:

   preorder: YEBAMDILRXTSOU
   inorder: ABMEIDYXRTLOSU

   (a) Create the binary tree, in other words print the binary tree that is represented by the above two traversals.
   (b) List the postorder traversal of the tree.

9. Consider the following list of numbers (input from left-to-right):

   $$13\ 9\ 5\ 7\ 12\ 2\ 3\ 14\ 6\ 1\ 10\ 15\ 8$$

   Create the 2-3 tree, showing **all** intermediate steps, in other words print the binary tree step-by-step.

10. Consider the following list of numbers (input from left-to-right):

    $$1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10$$

    (a) Create the heap, showing **all** intermediate steps, in other words print the binary tree step-by-step.
    (b) Show the heap's array representation.

11. Sort the following list of numbers using the Shell Sort (with $k = 3$). Show **all** work:

    $$13 \quad 9 \quad 5 \quad 7 \quad 12 \quad 2 \quad 3 \quad 14 \quad 6 \quad 1$$

12. Sort the following list of numbers using the quicksort. Show **all** work:

    $$13 \quad 9 \quad 5 \quad 7 \quad 12 \quad 2 \quad 3 \quad 14 \quad 6 \quad 1$$

13. Sort the following list of numbers using the heap sort. Use the array representation of the heap. Show **all** work:

    $$13 \quad 9 \quad 5 \quad 7 \quad 12 \quad 2 \quad 3 \quad 14 \quad 6 \quad 1$$

7.

a)

```
    n
m
        l
    i
            g
                f
        e
                c
            b
                a
```

b) m i e b a c g f l n

c) a b c e f g i l m n

8.

0 1 2 3 3 2 3 1 2 3 3 2 3 3
y e b a m d i l r x t s o u

a b m e i d y x r t l o s u
3 2 3 1 3 2 0 3 2 3 1 3 2 3

a)

```
      Y
   E    |    L
 B    D  |  R    S
A  M  I   |X  T  O  U
      |
```

b) AMBIDEXTROUSLY

9.

```
13
------------------------------------------------------------------------------------------
9,13
------------------------------------------------------------------------------------------
5,9,13


 9
5  13
------------------------------------------------------------------------------------------
  9
5,7  13
------------------------------------------------------------------------------------------
  9
5,7  12,13
------------------------------------------------------------------------------------------
  9
2,5,7  12,13


  5,9
2    7  12,13
------------------------------------------------------------------------------------------
  5,9
2,3   7  12,13
------------------------------------------------------------------------------------------
  5,9
2,3   7  12,13,14


  5,9,13
2,3    7,12  14


     9
  5      13
2,3   7   12    14
------------------------------------------------------------------------------------------
     9
  5      13
2,3   6,7   12    14
------------------------------------------------------------------------------------------
        9
  5     13
1,2,3  6,7   12  14


        9
 2,5     13
1  3 6,7   12  14
------------------------------------------------------------------------------------------
        9
 2,5     13
1  3 6,7   10,12  14
------------------------------------------------------------------------------------------
        9
 2,5     13
1  3 6,7   10,12  14,15
------------------------------------------------------------------------------------------
        9
 2,5     13
1  3 6,7,8   10,12  14,15


        9
 2,5,7     13
1  3,6 8   10,12  14,15


        5,9
  2        7   13
1 3,6 8   10,12  14,15
```