- **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)`.

   (b) Find `F(10)`.

   (c) Suppose + is changed to $\star$ in the inductive step. Now find `F(5)`.

   (d) What happens with the function call `F(-1)`?

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);
}
```

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

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

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

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

3. Draw each move for hanoi with 5 discs from tower 1 to tower 3. How many moves are required for 5 discs?
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., $86,401\text{s} = 1\text{day}, 1\text{sec}$).
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?
   (e) What is the complexity of Hanoi (in "big-o" terms)?
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.
   (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.
   (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.

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.
    (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 9 5 7 12 2 3 14 6 1

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

    13 9 5 7 12 2 3 14 6 1

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

    13 9 5 7 12 2 3 14 6 1