2016 Fall EE205

**Project #1 (Small Project, 50 pts):**

**C Practice with Hanoi Tower Puzzle**

**Due date: Sep. 28th (Wed), 11:59 PM (project #1)**

**TA: Kyounghye Kim (khye.kim@kaist.ac.kr)**

**Make sure understand clearly, what you do in project#1 by carefully reading. If you have some unclear things, ask your questions on the Q&A mail to TAs including CC to me (rhee.jk@kaist.ac.kr).**

**Ⅰ. Introduction**

Students are required to practice C coding in the Linux environment with the Hanoi tower puzzle.

**Ⅱ. Goal**

To present solutions of Hanoi tower solutions step by step on a text mode terminal in Linux. Write two solvers in iterative and recursive algorithms to produce exactly the same results. To understand and discuss the run time cost of the algorithm using a big O notation.

**Ⅲ. Ground rules**

Your program must be implemented by using C language and runnable on the provided Linux machines. You are not allowed to use any generics of C. In this project, you can use only the following header files**.**

*stdio.h, stdlib.h, string.h stdbool.h, stddef.h, stdint.h, sys/time.h*

**IV. Honoi Tower Puzzle**

Cf: <http://en.wikipedia.org/wiki/Tower_of_Hanoi>



**Object:**  to move the entire stack to another rod

**Rules for moves:**

1. Only one disk can be moved at a time.

2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.

3. No disk may be placed on top of a smaller disk.

**Labeling and presentation conventions in the project:**

Each disk is labeled with 1, 2, … N, in the smallest-to-largest disk order, where n is the count of disks. The tower positions are labeled with ‘a’, ‘b’, and ‘c’.

The solution for N =2 will be like

1: Disk 1 from a to b

2: Disk 2 from a to c

3: Disk 1 from b to c

**V. Requirements for implementing hanoi .**

The hanoi will read an input text file, which has a list of integers for the height N of the tower, and then, find soultions for each N, to generate output\_file s. It must be executed using the following command:

./hanoi *input\_file > output\_file*

#### VI. input\_file

The *input\_file* should be like following.

Heights of Honoi towers:

2

5

10

20

(You have to fscanf the first string and read in the integers.)

**VII.** **output\_file**

The *output\_file* should be like following.

N=2,

1: Disk 1 from a to b

2: Disk 2 from a to c

3: Disk 1 from b to c

Run time for N=2: *run time*

N=5,

**VIII. Analyses**

On a separate report, discuss analyses of runtimes for both algorithm and compare that with the results VII.

**IX. Submission**

Compress all files(Makefile, hanoi.c, hanoi.h, main.c, Report) into ‘<Student Number>\_Proj1.zip (xxxxxxxx\_Proj1.zip) and submit the zip file to KLMS.

**X. Delay penalty**

- 90% within 4 hours

- 80% within 8 hours

- 70% within 12 hours

- 60% within 24 hours

- 50% within 48 hours

- 0% after 48 hours