

CSOC 2430: HW4

Your Task:

For this homework, you will write a program to solve the puzzle of Tower of Hanoi. To learn about the puzzle, go to the link below.

[Wikipedia - Tower of Hanoi](#)

Tower of Hanoi can be solved using a recursive algorithm, to learn about the algorithm, you can read about it in Wikipedia or visit the link below.

[Writing a Towers of Hanoi program](#)

Traditionally Tower of Hanoi is played on 3 pegs with any number of disks. To make this homework more challenging (and fun), we will test your program with 3, 4 and 5 pegs with any number of disk.

- All cases with 3 pegs: 40 points
- All cases with 4 pegs: 40 points
- All cases with 5 pegs: 10 points
- Your code is elegant and follows the assignment specification: 10 points

Implementation Detail:

The goal is to move all disks from the first peg (left) to the very last peg (right).

You can assume at the beginning all disks are stacked at the first peg. Also the disks are diametrically sorted with the largest disk at the bottom and the smallest disk at top of peg.

Disks should be named using numbers, 1 is the smallest disk, 'n' is the largest disk. Pegs should be named using letters - the first peg 'A', then the second peg 'B' and so on. For 4 peg version of the problem, it involves moving all n disks from peg A to peg D using pegs C and B as intermediary pegs. For 5 peg version of the problem, it involves moving all n disks from peg A to peg E using pegs B, C and D as intermediary pegs.

The solution needs to be recursive and supports any number of disks.

Input

The program takes 3 arguments. **Number of Pegs**, **Number of Disks** and **Output File Name**.

For example, the following command asks for the solution of puzzle with 3 pegs, 4 disks, and the file will be written

to `3_4.txt` :

```
$ java HW4 3 4 "3_4.txt"
```

Output

Each line of the output file will be a single move of a single disk, with the format of `[DISK] [FROM] [TO]` . So disk 1 moved from peg A to peg B will be `1 A B` .

Below is the solution of 3 pegs and 3 disks, moving from peg A to peg C.

```
1 A C
2 A B
1 C B
3 A C
1 B A
2 B C
1 A C
```

Other

You need to at least uses all the pegs in order to claim the points for the corresponding cases. In other words, you will not get points if you solve the cases with 4 pegs but never used one of them.

Once again because the solution is not unique, we will not provide you with testing scripts. You should nonetheless be able to manually verify your solution with a pen and paper.

Run the program on Linux:

Create a directory on the Linux server, **its name must be hw4**

```
$ mkdir hw4
```

Change your current directory to the hw2

```
$ cd hw4
```

Run the shell script to compile the program

```
$ javac *.java
```