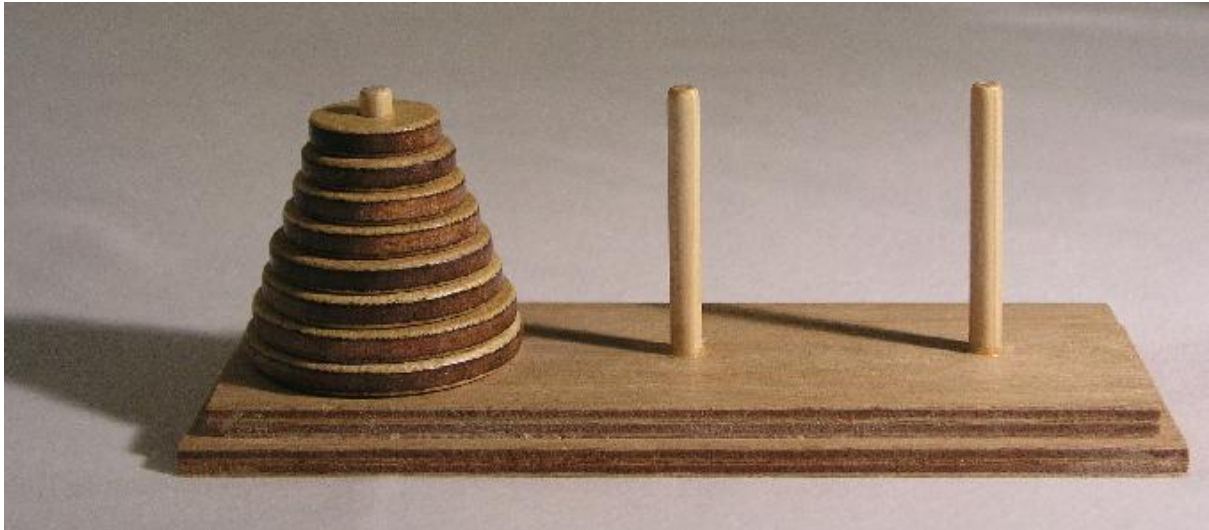


W4: Towers of Hanoi

Towers of Hanoi is a simple puzzle. See https://en.wikipedia.org/wiki/Tower_of_Hanoi for details.



Basically, there are three pegs. On the leftmost one sits a pile of disks, the biggest at the bottom and the smallest on top. The goal is to move the pile from the leftmost peg to the rightmost one, under the following constraints:

- You can only move one disk at a time.
- A bigger disk must never sit on top of a smaller one.

Write a program called `hanoi.cpp` that asks the user about how many disks the tower should consist of. (The height of the tower is called H .) Write a recursive function that solves the puzzle. (But first think of a condition with which to stop the recursion.) We call the three pegs, from left to right, *A*, *B*, and *C*. *A* is the *source* (where the tower sits initially). *C* is the *target* (where the tower shall be at the end), and *B* is a *helper* peg for storing disks on the go. The disks are numbered from 1 to H , with the biggest one at the bottom having number H .

The overall strategy for moving a tower of height H from *source* to *target* is as follows: (the roles of *source*, *helper*, and *target* change on different levels of the recursion)

1. Move the (upper part) tower of height $H-1$ from *source* to *helper*.
2. Move disk H (the lowest one) from *source* to *target*.
3. Move the tower of height $H-1$ from *helper* to *target*.

Your function should print the moves made right onto `cout`.

Correct runs of your program look as follows:

How many disks should the tower consist of? x
invalid input.

How many disks should the tower consist of? -2
invalid input.

How many disks should the tower consist of? 3
Move disk 1 from A to C.
Move disk 2 from A to B.
Move disk 1 from C to B.
Move disk 3 from A to C.
Move disk 1 from B to A.
Move disk 2 from B to C.
Move disk 1 from A to C.

Solve alone or in pairs

You can choose to solve a warm-up assignment all by yourself. But you may also choose to do it together with another student from the group. If so, one student must create a group (in CodeGrade) and the other student must join it.