T-Shirt Explanation: Westmont's 33rd Annual Mathematics Field Day

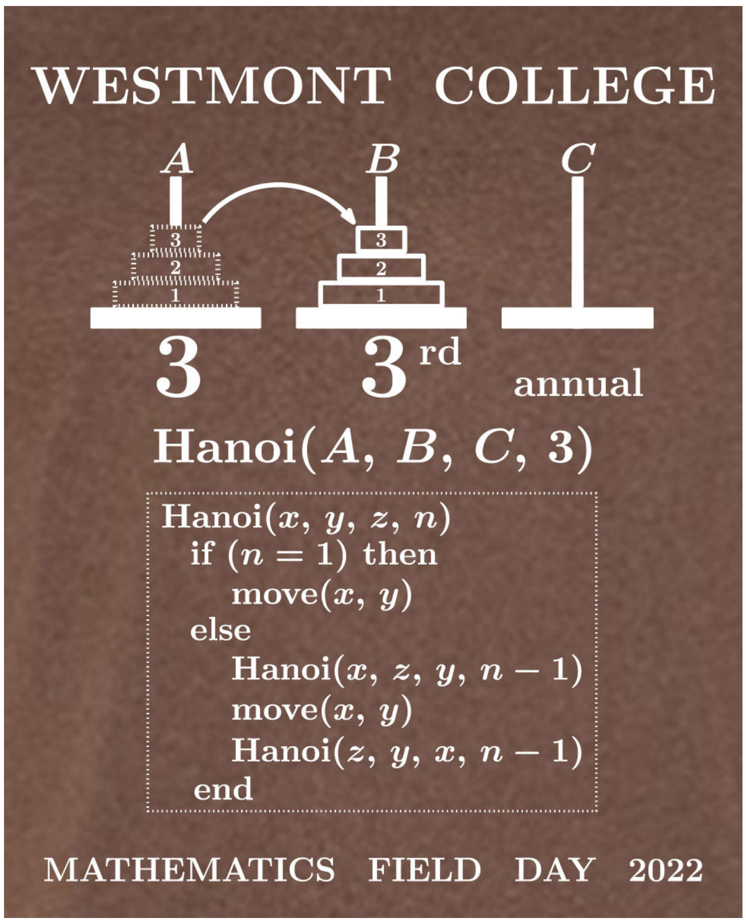
With a chalk talk topic of *Recurrence Relations*, the T-shirt for this year's event exhibits a recursive program that solves the "Tower of Hanoi" puzzle. The object of this puzzle is to move n discs resting on a peg onto an adjacent peg subject to the following rules:

1. Only one disc may be moved at a time.

2. Only a disc from the top of a stack of discs may be moved.

3. No disc may be place on top of a smaller disc.

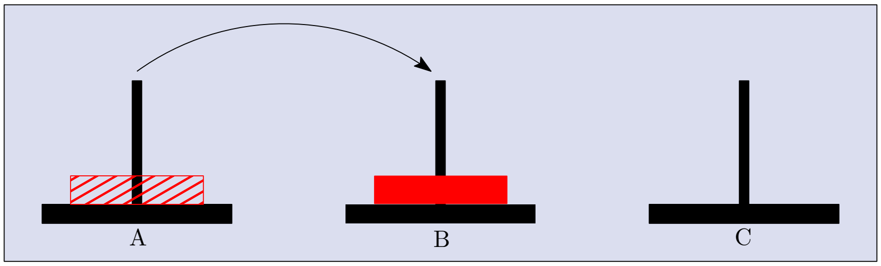
As the T-shirt illustrates, the computer program Hanoi(x, y, z, n) transfers three discs from peg A to peg B when it is invoked with the parameters $x = A$, $y = B$, $z = C$, and $n = 3$. The fact that *three* discs are moved connects with the 33rd annual event. Okay, that's a stretch, but it works!



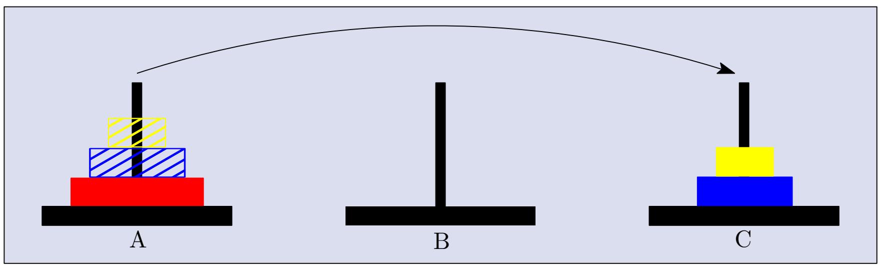
The program displayed on the T-shirt is written in pseudo-code, but it can easily be modified to suit any programming language that supports recursion. Understanding that it performs correctly involves the following principle, which is based on the strong form of mathematical induction:

If
a recursive procedure is correct when making no recursive calls,
and if
assuming that all prior recursive calls are correct *implies* that each program segment is correct,
then
the entire procedure is correct.

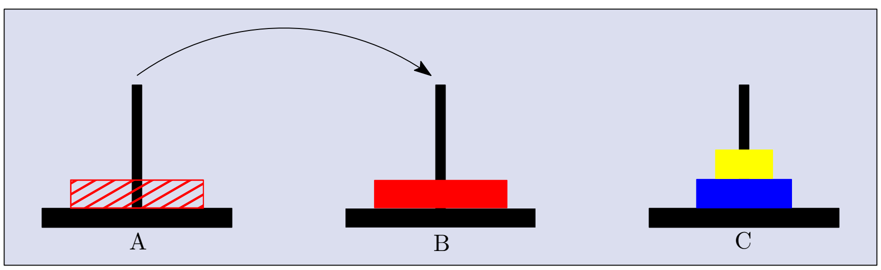
Let's see if that principle confirms the validity of the Hanoi program. To do so we assume that the function move(x, y) correctly transfers the top disc on peg x to peg y. It is clear that the only circumstance when no recursive calls are made occurs when $n = 1$, so in that case the move(x, y) function will transfer the top disc from peg $x = A$ to peg $y = B$, as shown below.



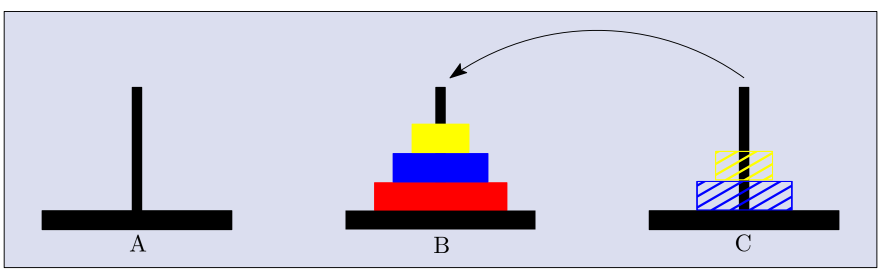
We illustrate recursive calls with $n = 3$ discs. The program first checks to see if $n = 1$. It is not, so the procedure then makes two recursive calls. As the figure below indicates, the outcome of first recursive call, Hanoi$(x, z, y, n-1)$, is that $n - 1 = 2$ discs get moved from peg $x = A$ to peg $z = C$, with peg $y = B$ serving as an intermediary transfer location. The procedure does not move two disks at once, of course, but during its execution it recursively invokes its own code (*i.e.*, it calls itself), and in doing so follows all the rules in accordance with the "inductive" hypothesis.



Next, the move(x, y) function transfers the remaining disc on peg $x = A$ to peg $y = B$.



Finally, and following all the required rules again, the outcome of Hanoi$(z, y, x, n-1)$ is that the $n - 1 = 2$ discs on peg $z = C$ get moved to peg $y = B$, with peg $x = A$ serving as an intermediary transfer location.



For a look at a Java program implementing this computer code click the following link:
https://www.youtube.com/watchv=8TnMIGNJlb0.