

Overview:	Find the number of paths between three islands such that one can collect all the fruit.
Description:	<p>Baymax has learned the trick to cheering Hiro up: flying!! After discovering this on a joyride over San Fransokyo, he and the rest of Hiro's friends from the lab decide to incorporate flying into their next "Happy Hiro" project. They set a basket of <math>n</math> cherries atop each of 3 different hot air balloons. Hiro naturally wants all the fruit at once, but to keep him occupied and flying more, they rig the balloons so that if he tries to take more than one at a time, the balloon will pop and send the fruit falling to the ground.</p> <p>As an example, suppose our three balloons are A, B, and C and Hiro collects a cherry from island A. He cannot collect another cherry from A next then, but he <i>can</i> collect another cherry from island B and return to A afterwards to collect a piece of fruit.</p> <p>The science-loving Fred watches these antics and begins wondering: how many different paths between balloons can Hiro take to collect all <math>3n</math> cherries? These different paths should include different balloons to start from, and a path is over when the last cherry has been collected. Can you help him figure it out?</p>
Filename:	islands.{java, cpp, c, cc, py}
Input:	<p>The input will contain 1 line</p> <p>The first line is <math>n</math>, the number of cherries per balloon</p>
Output:	Output a single integer, the number of possible paths we can take to collect all the fruit.
Assumptions:	$1 \leq n \leq 10$
Sample Input #1:	1
Sample Output #1:	6
Sample Input #2:	2
Sample Output #2:	30