***“ …Write a program that takes both the initial size of a green crud population (in pounds) and a number of days as input and outputs the number of pounds of green crud after that many days. Assume that the population size is the same for four days and then increases every fifth day. Your program should allow the user to repeat this calculation as often as desired.”***

My thinking on solving this was to take the user’s ‘crud weight’ input and use it to populate an array that has increasing values in accordance with the Fibonacci sequence.

If the user’s crudInput is 0, or if their days input is < 10, then nothing changes, and the control goes to the else statement.

If the user’s input for the amount of crud > 0, and the number of days is >= 10, then we set the length of the array ‘growthSequence’ to equal 1 for the initial amount, plus the quotient of the amount of user input days divided by the number of days between each increase in population (5). In the case of 20 as the user’s input of the amount of days, int[] growthSequence = new int[1+(daysInput / 5)], which works out to an array length of 5.

Days 1-9 incur no growth, so array indices [0] and [1] are set to the user’s input. Array index [2] will be the sum of [0] and [1], [3] is the sum of [1] and [2], and so on.

growthSequence[0] and growthSequence[1] are set to equal the user’s ‘crud’ input, while a new int crudPostGrowth is initialized to zero for the purpose of storing the growth changes to print later.

A loop begins at growthSequence[2], where the index[2] is assigned the value of the sum of [i-1] and [i-2], the new value at index[2] is copied to the variable crudPostGrowth, and i is incremented by 1.

When int i == growthSequence.length, the loop terminates, and the string hasGrown, with the results of the loop, is printed.

All of this is wrapped in a do/while statement, which is dependent upon the input of the user at the end of the first run of the sequence. If the user wants to run again, it starts over. If not, a message is printed and the program ends. I think ideally I would determine whether to run again with a method that validates the user’s input, and I will learn more about this.



