

# Recursive Functions

Level 4++ (All done well)  
Level 4 (1-5 OR 1-4 and 6)  
Level 3 (1-4)  
Level 2 (1-3)

1) Write a program that generates first  $n$  terms of the recursive sequences given their definitions below. This may require more than one function working together. Remember to not use loops!

a)  $t_1 = 3$ ;  $t_n = t_{n-1} - 2$

b)  $t_1 = -1$ ;  $t_n = 2t_{n-1}$

2) Write a recursive function `gcd` that returns the greatest common divisor of  $x$  and  $y$ . The gcd of  $x$  and  $y$  is defined recursively as follows: If  $y$  is equal to 0, then  $\text{gcd}(x, y)$  is  $x$ ; otherwise  $\text{gcd}(x, y)$  is  $\text{gcd}(y, x\%y)$ , where  $\%$  is the modulus operator.

3) Write the following *recursive* functions to manipulate strings:

a) Create a function `reverse` that *returns* the reverse of a string.

b) Create a function `isPalindrome` that *returns* whether a string input is a palindrome.

4) Write a recursive function `decimalToBinary` that takes an integer input and *returns* the number in binary as a string.

5) Write a function that *recursively* finds the average of all values in an integer array. I would like you to avoid a method which involves finding the sum and dividing by the number of elements. *This will be trickier than it seems.*

6) Factorials

Marcia just learned about factorials in school. She understood that they are the product of all the integers between 1 and the number itself. For example,  $\text{factorial}(5) = 5 \times 4 \times 3 \times 2 \times 1 = 120$  and  $\text{factorial}(10) = 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 3,628,800$ .

She decided that these numbers are too big, and wasteful. So she invented fractorials. They are the smallest possible numbers that can be evenly divided by all the whole numbers from 1 to itself. So for example  $\text{fractorial}(5) = 60$ , since  $60/1$ ,  $60/2$ ,  $60/3$ ,  $60/4$ ,  $60/5$  are all whole numbers, something which is not true for any smaller number. Fractorial (10) incidentally is 2520. Something much easier to manage than 3,628,800!!!

The problem is, that they are not so easy to construct. Please help her, by writing a program that will calculate fractorials. *You may create any helper functions you feel are necessary.*

Sample Input:	Sample Output:
18	Fractorial (4) = 12
4	Fractorial (17) = 12252240
17	Fractorial (22) = 232792560
22	Fractorial (12) = 27720
12	Fractorial (1) = 1