Lab 4: Recursion Puzzles

In this lab, you will implement a series of recursive algorithms. All of your functions must be recursive. If your functions are not recursive, you will not get credit for completing this lab. Your functions must use the signatures I've provided below.

Your Functions

1. `int findMax(const int arr[], size_t size)` - Finds and returns the maximum value in an integer array of length *size*. Keep in mind that integers may be negative.
2. `double maxMult(const int arr[], size_t size, int mult)` - Finds and returns the maximum value in the integer array *arr* that is a multiple of *mult*, or NAN, defined as `0.0/0.0` if no match is found.
3. `bool canAdd(const double arr[], size_t size, double target)` - Returns true if there are two values that, when added, sum to *target*. For example, given the array $[1, 3, 5, 7]$ and target 10, the function should return true since $3 + 7 = 10$. Given the same array and target 20, the function should return false.
4. `bool isSubString(const std::string& word, const std::string& toFind)` - Returns true if the string *toFind* is a substring of *word*. Returns false otherwise. You may NOT use the string class's *find* type functions (this includes find, find_first_of, find_last_of, find_first_not_of, find_last_not_of, rfind)

Check list

Your code will be run through unit tests. The check list does not apply for this lab.

Example Output

While you probably want a main method to test your functions, there is no specific way you must set up this method. Your functions will be tested independently.

What to Turn In

Upload the .cpp and .h files that were provided, filled in with your code. Do not change the function signatures!! Make sure you submit all of the files in one submission.