Introduction to Programming

Labs – Week 10b

Exercise 1

Write a recursive function `isPalindrome()` that returns `true` if and only if the input string is palindrome (a palindrome is a word, number, phrase, or other sequence of characters which reads the same backward as forward, such as `"Madam"` or `"Racecar"` or the number `"10201"`.)

Hint: Ensure that the first and the last letters are the same, and then recursively check the rest of the string.

Exercise 2

Write a recursive method to find maximum elements in a given array. Implement the following recursive function.

```
public static int findMax(int[] a)
   { return findMax(a, a.length; }
// This helper function finds the maximum of first n elements in the array
public static int findMax(int[] a, int n) {
   if(n==____) return _____; // base case
   int m = _________; // recursively find the maximum of n-1 elements
   return _____________; // return maximum of m and the last element
}
```

Exercise 3

Write a recursive method `sumTo` that accepts an integer parameter `n` and returns the sum of the first `n` reciprocals. In other words: `sumTo(n)` returns: `1 + 1/2 + 1/3 + 1/4 + ... + 1/n.` Assume that your method is always passed a value greater than equal to `0`.

Exercise 4

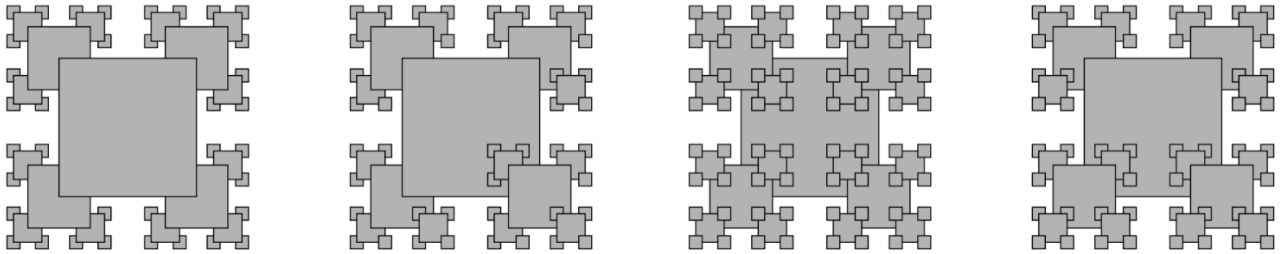
Write a recursive method `writeChars` that accepts an integer parameter `n` and that prints out `n` characters as follows. The middle character of the output should always be an asterisk ("`*`"). If you are asked to write out an even number of characters, then there will be two asterisks in the middle ("`**`"). Before the asterisk(s) you should write out less-than characters ("`<`"). After the asterisk(s) you should write out greater-than characters ("`>`"). For example, the following calls produce the following output:

Call	Output
`writeChars(1)`	`*`
`writeChars(2)`	`**`
`writeChars(3)`	`<*>`
`writeChars(4)`	`<**>`
`writeChars(5)`	`<<*>>`
`writeChars(6)`	`<<**>>`
`writeChars(7)`	`<<<*>>>`
`writeChars(8)`	`<<<**>>>`

Assume that your method is always passed a value greater than equal to `1`.

Exercise 5

Write a program to produce each of the following recursive patterns. The ratio of the sizes of the squares is `2.2:1`. To draw a shaded square, draw a filled gray square, then an unfilled black square.



Exercise 6

Given a `9`-by-`9` array of integers between `1` and `9`, write `SudokuCheck.java` to check if it is a valid solution to a Sudoku puzzle: each row, column, and block should contain the 9 integers exactly once.

```
5 3 4 | 6 7 8 | 9 1 2
6 7 2 | 1 9 5 | 3 4 8
1 9 8 | 3 4 2 | 5 6 7
------+-------+------
8 5 9 | 7 6 1 | 4 2 3
4 2 6 | 8 5 3 | 7 9 1
7 1 3 | 9 2 4 | 8 5 6
------+-------+------
9 6 1 | 5 3 7 | 2 8 4
2 8 7 | 4 1 9 | 6 3 5
3 4 5 | 2 8 6 | 1 7 9
```

The above Sudoku board can be stored in a 2d array as follows:

```
int[][] a = {
{5, 3, 4, 6, 7, 8, 9, 1, 2},
{6, 7, 2, 1, 9, 5, 3, 4, 8},
{1, 9, 8, 3, 4, 2, 5, 6, 7},
{8, 5, 9, 7, 6, 1, 4, 2, 3},
{4, 2, 6, 8, 5, 3, 7, 9, 1},
{7, 1, 3, 9, 2, 4, 8, 5, 6},
{9, 6, 1, 5, 3, 7, 2, 8, 4},
{2, 8, 7, 4, 1, 9, 6, 3, 5},
{3, 4, 5, 2, 8, 6, 1, 7, 9} };
```