Выполнить предложенные ниже задания, не используя изменяемые переменные (var). Где необходимо, использовать хвостовую рекурсию.

#### Задание №1: Треугольник Паскаля

Треугольником Паскаля называют фигуру из чисел следующего вида

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

...

Нумерация строк и столбцов начинается с нуля, причем первым (нулевым) элементом каждой строки считается ее крайний левый элемент.

Описать функцию

def pascal(c: Int, r: Int): Int

, принимающую на вход номер столбца (параметр «*c*») и строки (параметр «*r*»), и возвращающую значение числа, расположенного в соответствующей позиции треугольника Паскаля. Например, «*pascal(0, 2)≡ 1*», а «pascal(1, 3) *≡ 3*».

#### Задание №2: Баланс скобок

Напишите рекурсивную функцию

def balance(chars: List[Char]): Boolean

, проверяющую корректность баланса круглых скобок во входной строке. Учтите, что при реализации необходимо проверить не только совпадение открывающих и закрывающих скобок по количеству, но и их корректную последовательность по отношению друг к другу. Например,

balance("(if (zero? x) max (/ 1 x))".toList)≡ true

balance("((x)(y)((z)+k)-f)".toList) ≡ true

balance(")(".toList) ≡ false

balance("(()".toList) ≡ false

#### Задание №3: Помогите кассиру

Сотруднику одной распространенной американской закусочной необходимо как можно быстрее освободить кассу. Для этого он очень хочет подсчитать, сколькими способами он может дать сдачу определенного размера, если у него на руках есть только монеты заданного достоинства.

Напишите функцию

def countChange(money: Int, coins: List[Int]): Int

, принимающую на вход размер сдачи, список достоинств (типов) монет и возвращающую количество способов, которыми сотрудник может дать сдачу. Считать, что у кассира есть неограниченное количество монет каждого типа. Например,

countChange(4, List(1, 2)) ≡ 3 //1+1+1+1; 1+1+2; 2+2