

1	<p>Закаренная функция</p> <pre>arith: (Int => Boolean) => (Int, Int, Int) => List[Int],</pre> <p>возвращающая список из n первых членов арифметической прогрессии с начальным членом a_0 и разностью d, удовлетворяющих некоторому предикату.</p>
2	<p>Закаренная функция comb: $\text{Int} \Rightarrow (\text{List}[\text{Int}] \Rightarrow \text{List}[\text{List}[\text{Int}]])$, формирующая список всех сочетаний элементов списка целых чисел. Размер сочетания передаётся через параметр функции.</p>
3	<p>Функция coprimes: $\text{List}[\text{Int}] \Rightarrow \text{List}[(\text{Int}, \text{Int})]$, выполняющая поиск в списке целых чисел пар взаимно простых чисел. Функция должна возвращать список найденных пар, причём в каждой паре первое число должно быть меньше второго.</p>
4	<p>Закаренная функция digits: $\text{Int} \Rightarrow (\text{Int} \Rightarrow \text{List}[\text{Int}])$, выполняющая перевод числа в заданную систему счисления (параметр функции – основание системы счисления).</p>
5	<p>Функция fib: $(\text{Int}, \text{Int} \Rightarrow \text{Boolean}) \Rightarrow \text{List}[\text{Int}]$, порождающая последовательность чисел Фибоначчи, не превышающих заданного целого числа и удовлетворяющих некоторому предикату.</p>
6	<p>Функция filterIndexes: $(\text{List}[\text{Int}], \text{Int} \Rightarrow \text{Boolean}) \Rightarrow \text{List}[\text{Int}]$, удаляющая из списка целых чисел те числа, номера которых в списке не удовлетворяют заданному предикату.</p>
7	<p>Функция flatten: $(\text{List}[\text{List}[\text{Int}]], \text{Int} \Rightarrow \text{Boolean}) \Rightarrow \text{List}[\text{Int}]$, выполняющая конкатенацию списков целых чисел, находящихся в списке списков целых чисел и имеющих длину, удовлетворяющую предикату.</p>
8	<p>Функция fractions: $\text{List}[\text{Int}] \Rightarrow \text{List}[(\text{Int}, \text{Int})]$, по последовательности целочисленных коэффициентов $a_0, a_1, a_2, \dots, a_n$ вычисляющая последовательность конечных цепных дробей:</p> $a_0, a_0 + \frac{1}{a_1}, a_0 + \frac{1}{a_1 + \frac{1}{a_2}}, a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3}}}, \dots$
9	<p>Закаренная функция frames: $\text{Int} \Rightarrow (\text{List}[\text{Int}] \Rightarrow \text{List}[\text{List}[\text{Int}]])$, формирующая список, состоящий из всех подсписков списка целых чисел указанной в качестве параметра функции длины. Подписком будем считать список, который можно получить удалением произвольного количества элементов от начала и от конца списка.</p>
10	<p>Функция kadane: $\text{List}[\text{Int}] \Rightarrow (\text{Int}, \text{Int})$, выполняющая поиск границ подпоследовательности с максимальной суммой (алгоритм Кадана).</p>

11	Функция <code>lists: List[Int] => List[(Int, Int)]</code> , принимающая отсортированный по возрастанию список целых чисел и выполняющая разбиение списка на непрерывные интервалы, в которых каждое следующее число на единицу больше предыдущего.
12	Функция <code>merge: (List[Int], List[Int]) => List[Int]</code> , выполняющая слияние двух отсортированных по возрастанию списков целых чисел в один отсортированный список.
13	Функция <code>mul: (List[Int], List[Int]) => List[Int]</code> , выполняющая умножение двух целых чисел, каждое из которых представлено списком степеней своих простых делителей.
14	Функция <code>pack: List[Int] => List[List[Int]]</code> , выполняющая упаковку последовательно идущих одинаковых элементов списка в подсписок (т.е., например, для списка <code>List(1,1,2,2,2,1)</code> функция должна возвращать <code>List(List(1,1),List(2,2,2),List(1))</code>).
15	Функция <code>palindromes: List[Int] => List[List[Int]]</code> , принимающая последовательность целых чисел и выполняющая поиск всех непрерывных подпоследовательностей, являющихся палиндромами.
16	Функция <code>partition: (List[Int], Int) => (List[Int], List[Int])</code> , разделяющая список целых чисел на два списка: в первый список помещаются числа, которые меньше указанного числа, а во второй – числа, которые не меньше.
17	Функция <code>partitionP: (List[Int], Int => Boolean) => (List[Int], List[Int])</code> , разделяющая элементы исходного списка на два списка в зависимости от того, удовлетворяют ли они предикату.
18	Функция <code>peaks: List[Int] => List[Int]</code> , формирующая список индексов пиков последовательности (пик – такой элемент, что соседние элементы его не превышают).
19	Закаренная функция <code>power: Int => (Int => Int)</code> , выполняющая быстрое возведение числа в указанную степень (параметр функции – степень).
20	Функция <code>powers: (List[Int], Int => Boolean) => List[Int]</code> , удаляющая из списка те числа, которые не являются заданными предикатом степенями числа 2.

21	Закаренная функция <code>repeat: Int => List[Int] => List[Int]</code> , формирующая список, состоящий из элементов исходного списка, каждый из которых повторён указанное количество раз.
22	Функция <code>reverse: List[List[Int]] => List[List[Int]]</code> , выполняющая переворачивание как самого списка списков, так и каждого из его элементов.
23	Функция <code>reverseP: (List[Int], Int => Boolean) => List[Int]</code> , выполняющая переворачивание списка целых чисел и удаление из него элементов, не удовлетворяющих предикату.
24	Закаренная функция <code>slices: Int => (List[Int] => List[List[Int]])</code> , выполняющая разбиение списка целых чисел на фрагменты указанной в качестве параметра функции длины.
25	Закаренная функция <code>sorted: ((Int, Int) => Boolean) => (List[Int] => Boolean)</code> , принимающая функцию сравнения двух целых чисел и возвращающая функцию, определяющую, является ли список целых чисел отсортированным в соответствии с функцией сравнения.
26	Функция <code>sortedSeq: List[Int] => List[List[Int]]</code> , разбивающая список целых чисел на непересекающиеся подсписки, в пределах которых числа либо не возрастают, либо не убывают. Подписком будем считать список, который можно получить удалением произвольного количества элементов от начала и от конца списка.
27	Функция <code>split: (List[Int], Int => Boolean) => List[List[Int]]</code> , выполняющая разбиение последовательности целых чисел на подпоследовательности, разделённые числами, удовлетворяющими предикату.
28	Функция <code>sublists: (List[Int], Int) => List[List[Int]]</code> , разбивающая список целых чисел на непересекающиеся подсписки, сумма элементов которых не превышает указанного числа. Подписком будем считать список, который можно получить удалением произвольного количества элементов от начала и от конца списка.
29	Функция <code>sumDigits: (Int, Int => (Int, Int)) => Int</code> , выполняющая суммирование цифр числа (функция принимает замыкание, возвращающее младшую цифру числа и число, лишённое этой цифры).
30	Функция <code>trim: (List[Int], Int => Boolean) => List[Int]</code> , выполняющая удаление из списка подряд идущих нулей, количество которых удовлетворяет предикату.
31	Функция <code>uniq: List[Int] => (List[Int], Boolean)</code> , удаляющая дублирующиеся числа из списка отсортированных по возрастанию целых чисел. Функция возвращает пару, первым элементом которой является результирующий список, а вторым – признак успешного выполнения. Выполнение может быть неуспешным, если исходный список не отсортирован по возрастанию.