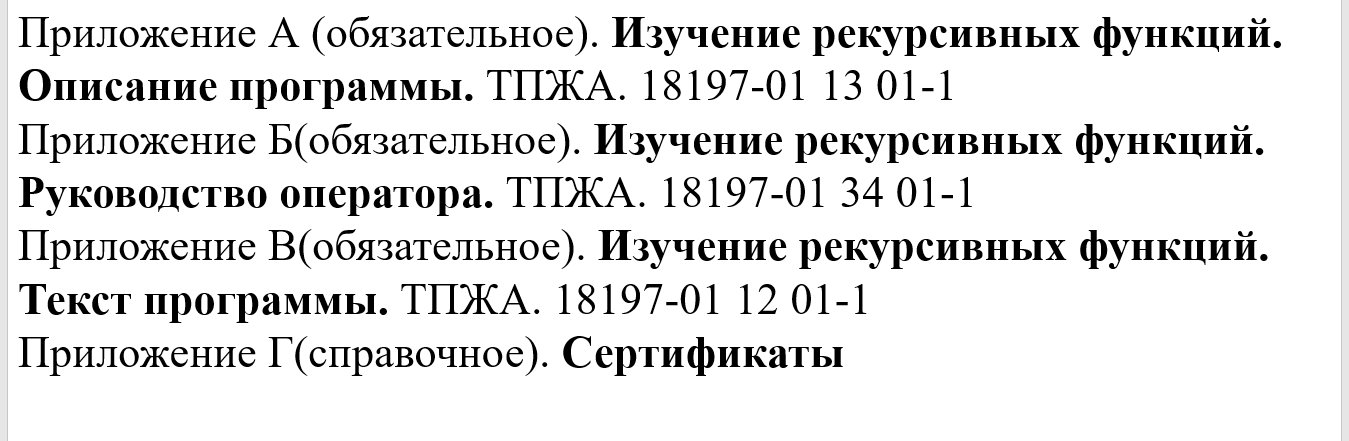
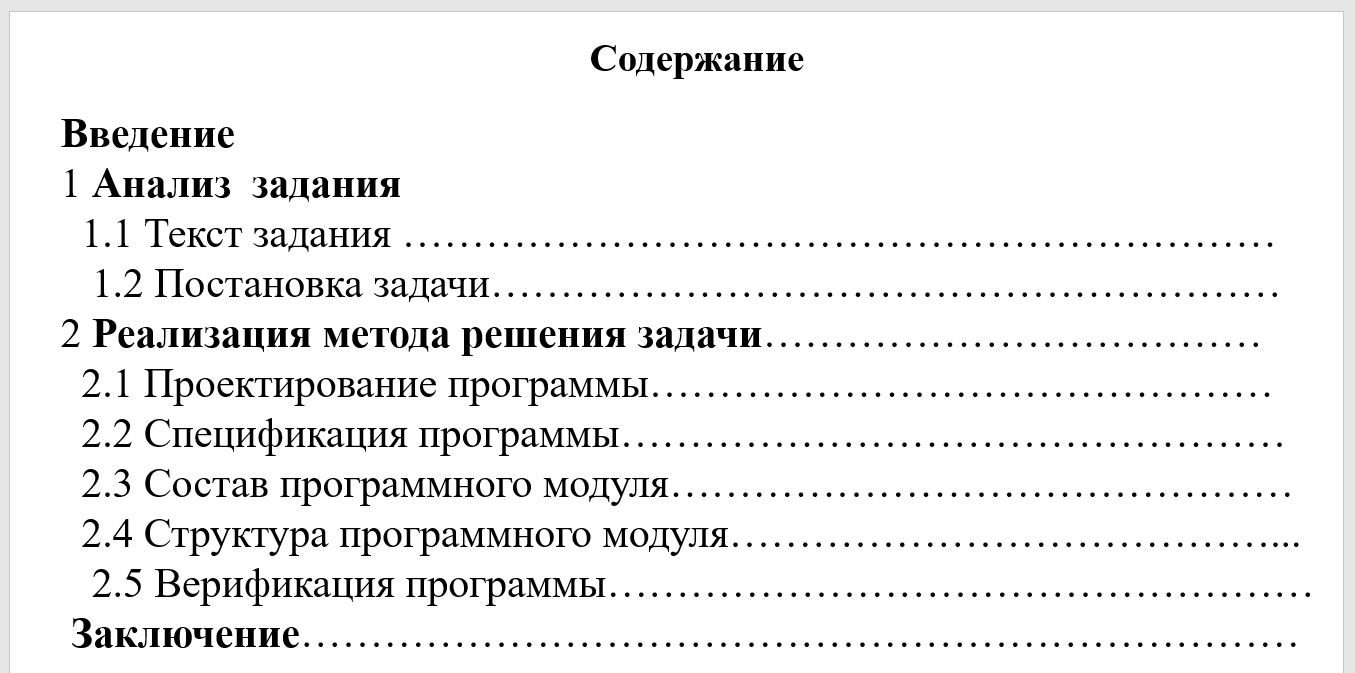
**Требования к заданию “Программирование   
рекурсивных функций” и   
ВАРИАНТЫ**

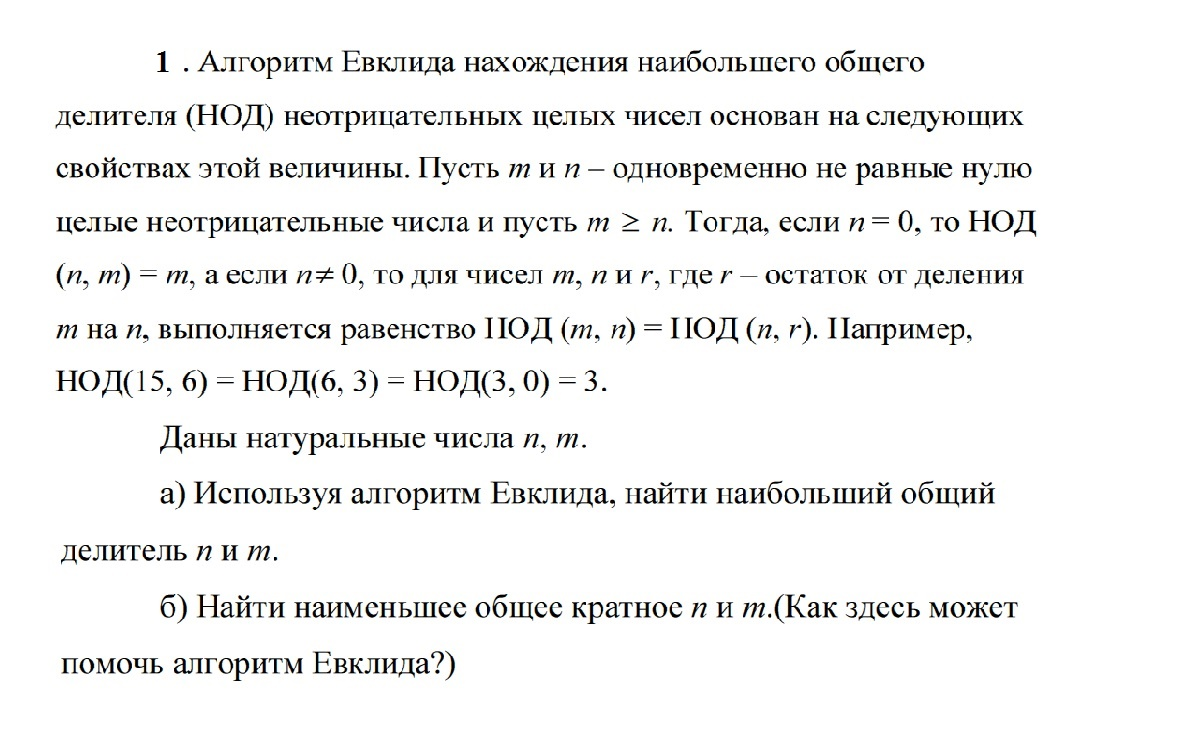
Для решения этой задачи разработать следующие функции:

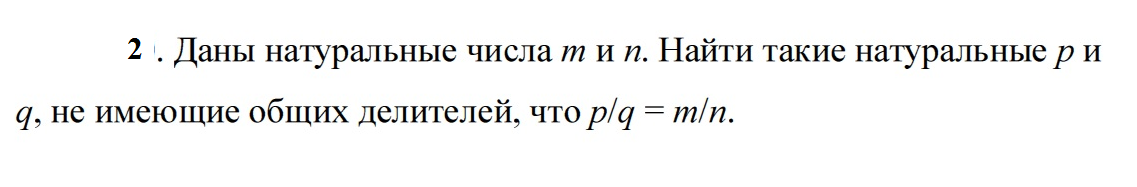
- функцию **ввода**;  
- функцию, вычисляющую выражение **рекурсивным способом**;  
- функцию, осуществляющую **проверку вычислений с помощью цикла**;  
- функцию **main()**, которая **организует работу** всех разработанных функций; в этой функции создать файлы для записи и чтения.   
 Предусмотреть эхо-контроль введенных данных и защиту от некорректного ввода. В программе должно быть 3 файла:  
- **файл с головной программой**, откуда вызываются все остальные функции;  
- **заголовочный файл** с прототипами функций;

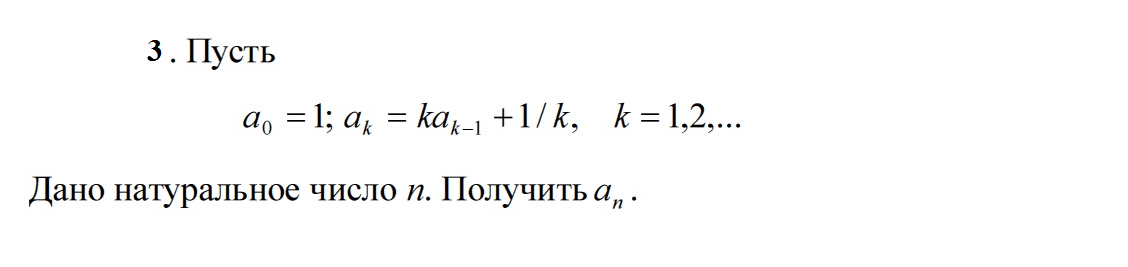
- **файл с кодами всех функций**. Имя заголовочного файла и файла с исходниками функций должно совпадать, например **my.h** и **my.cpp**

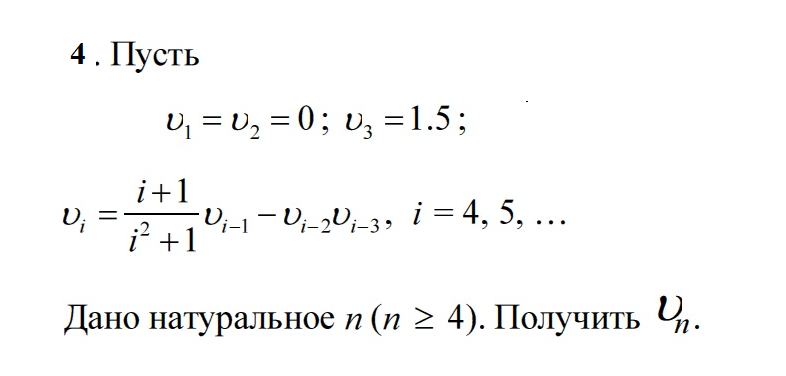


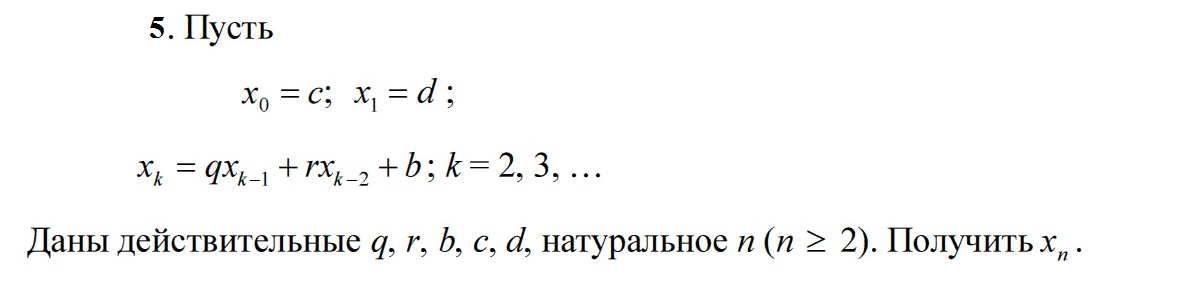
**СО СЛЕДУЮЩЕЙ СТРАНИЦЫ ВАРИАНТЫ**

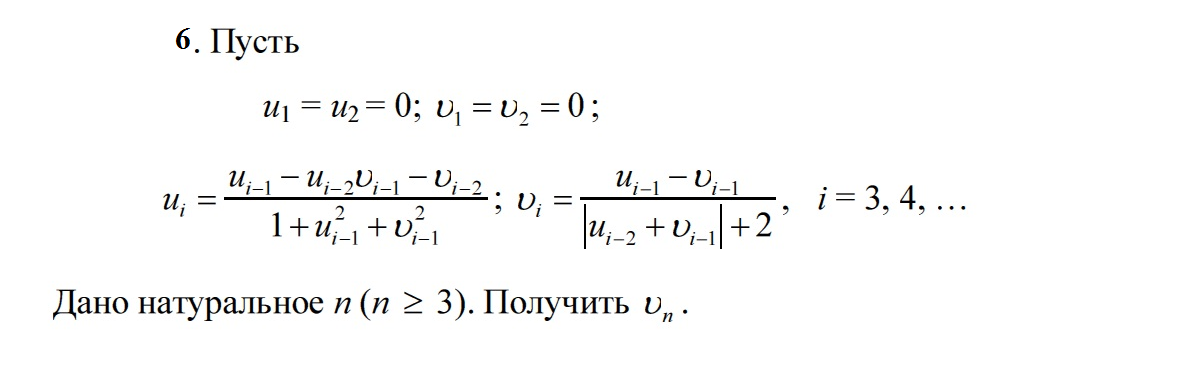


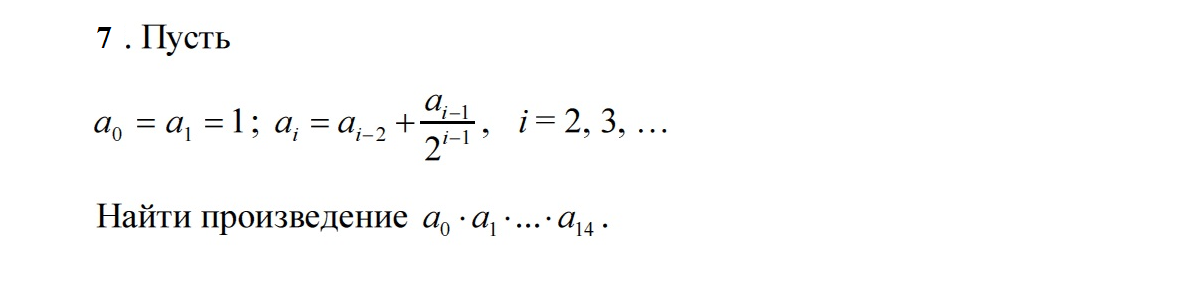


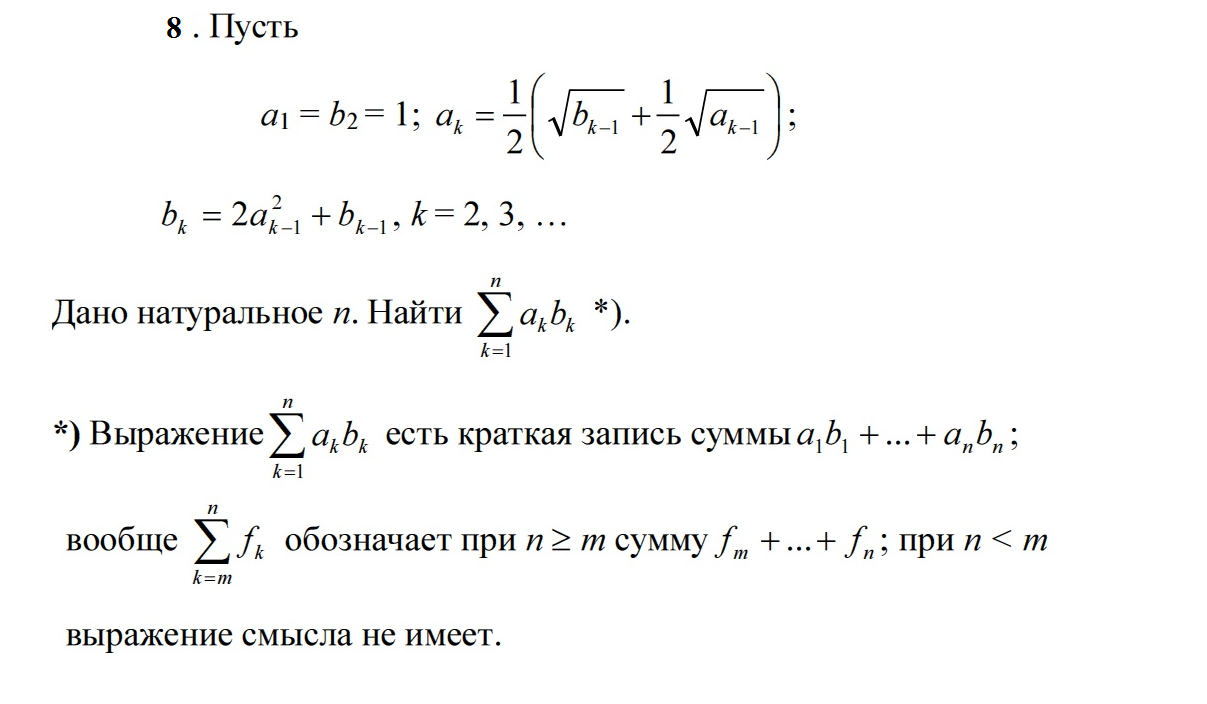


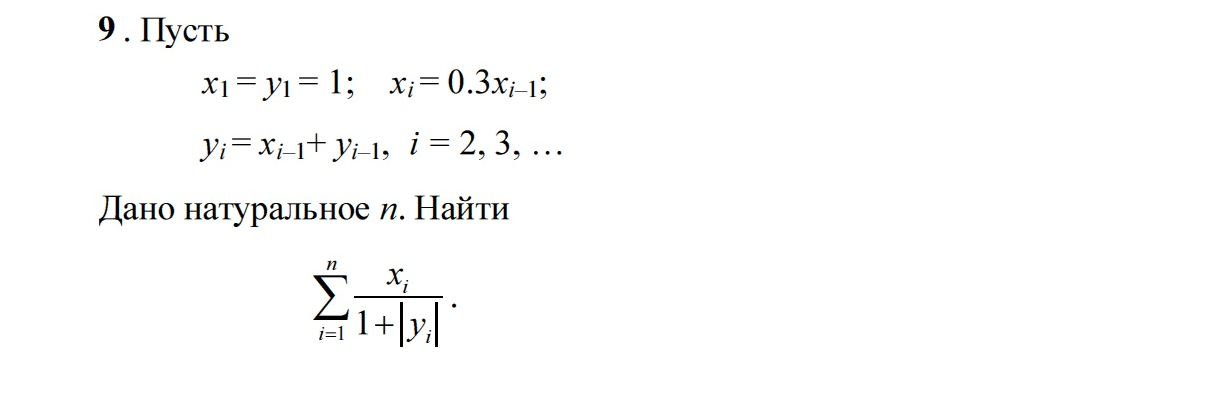


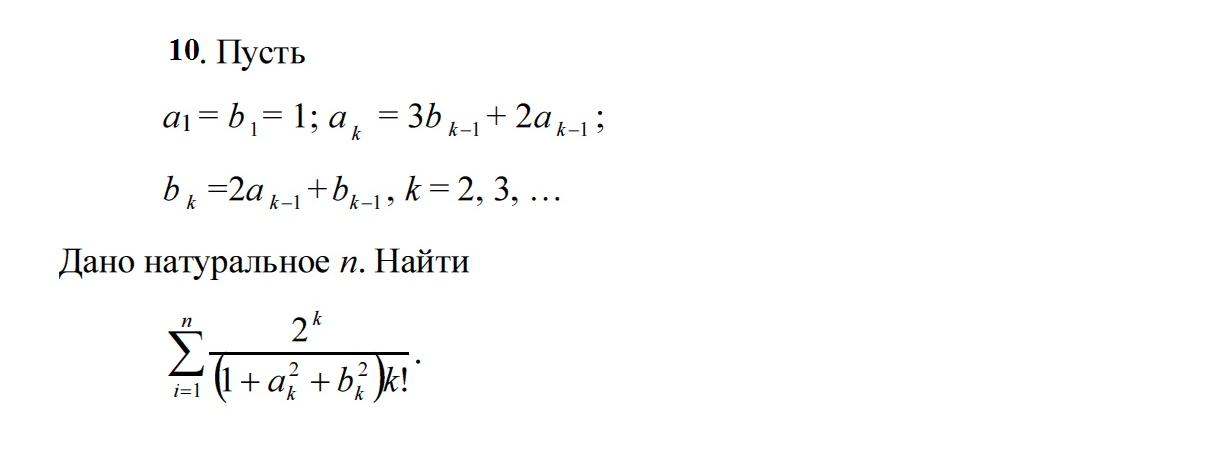


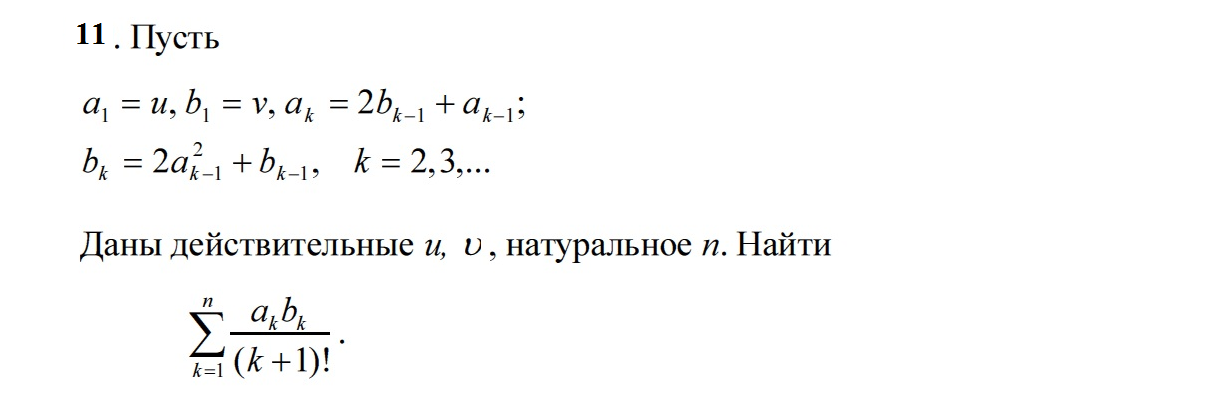


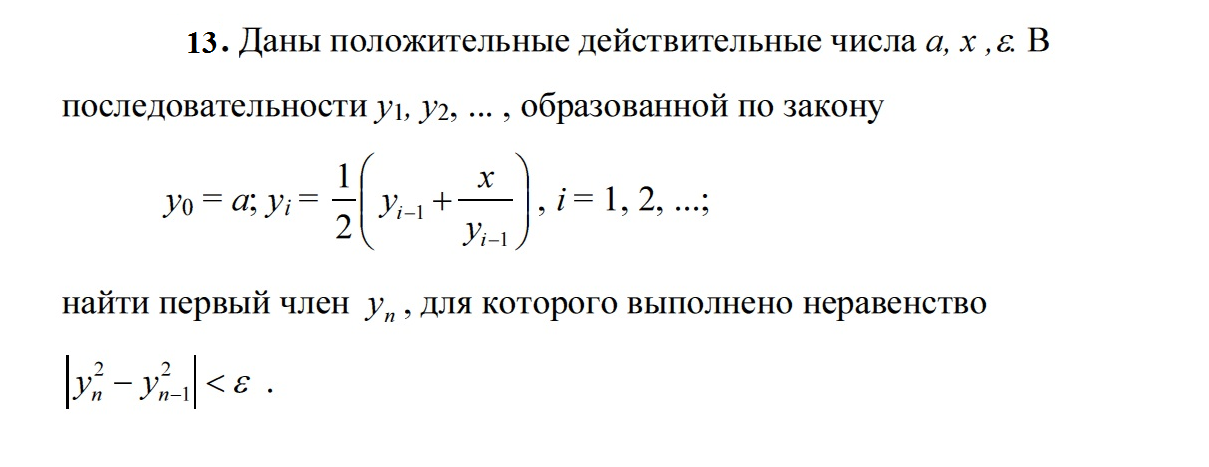
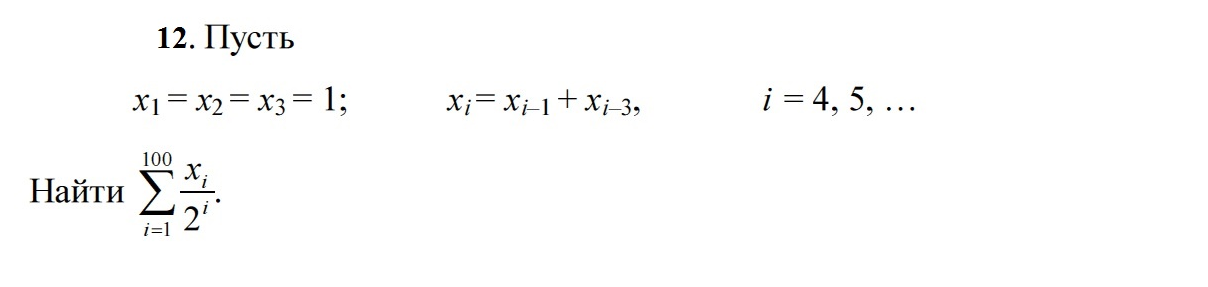


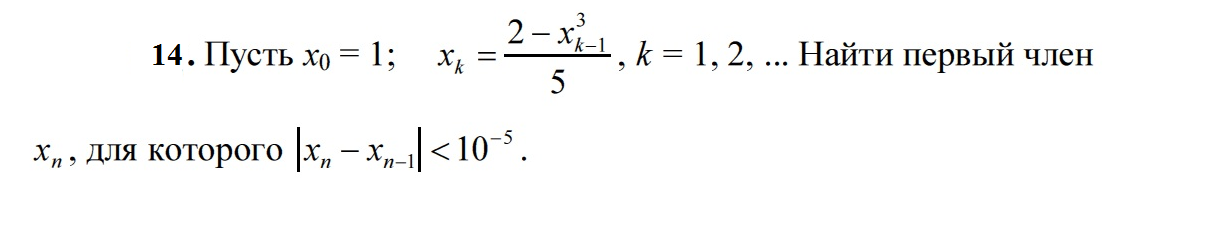


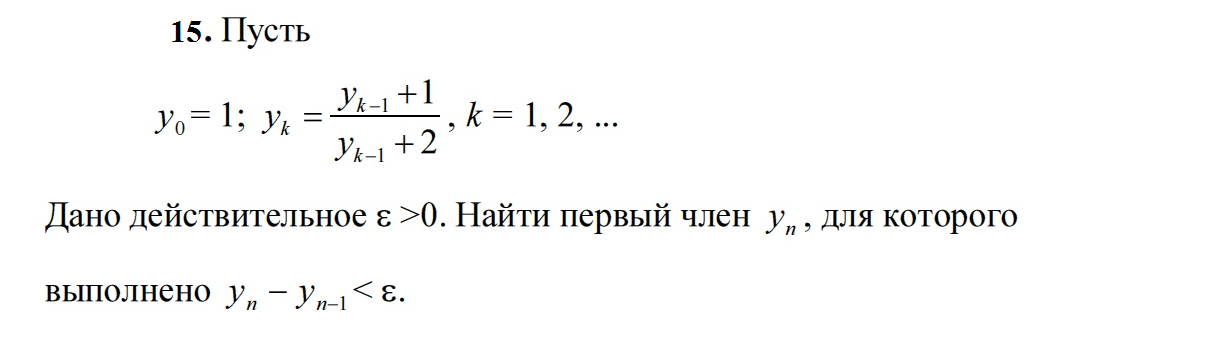


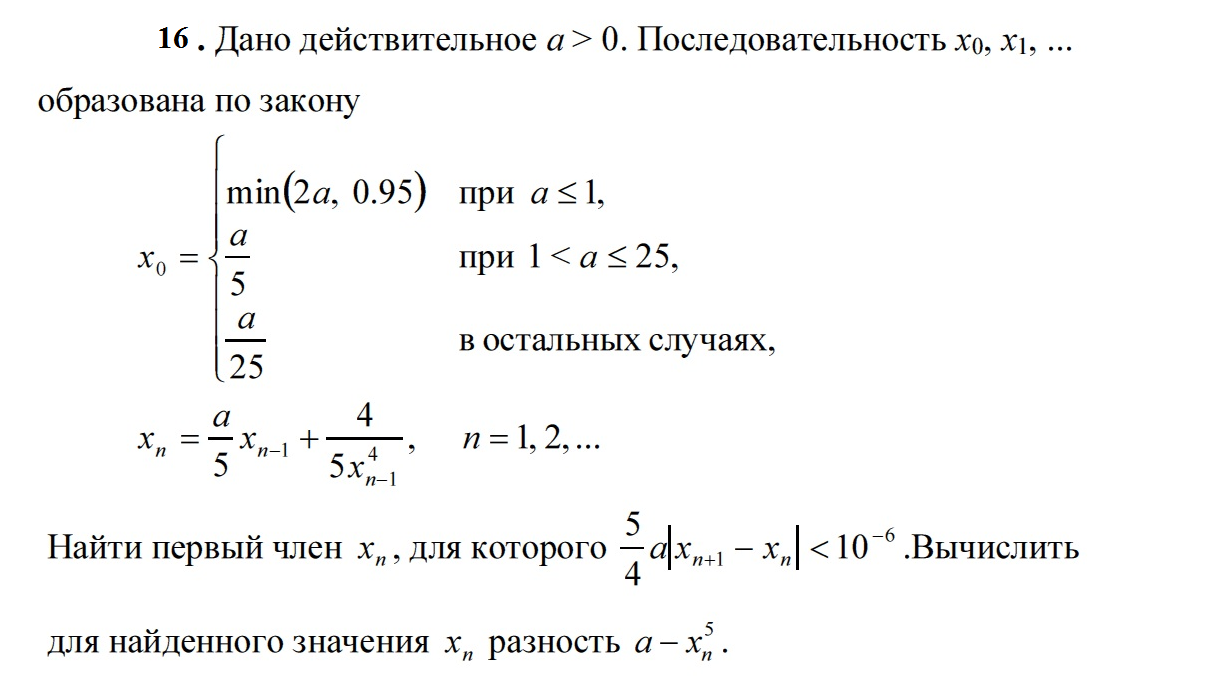


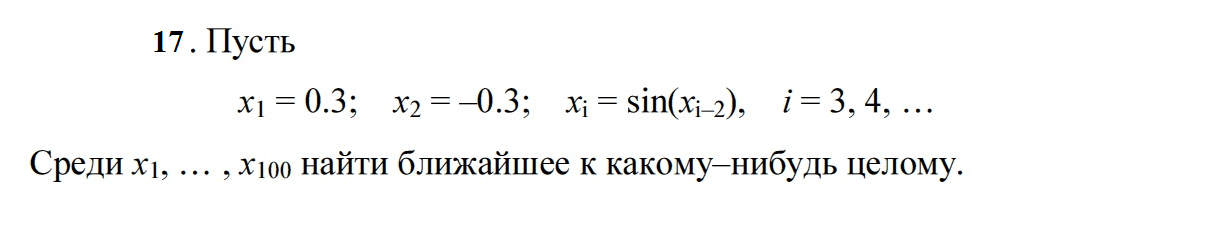


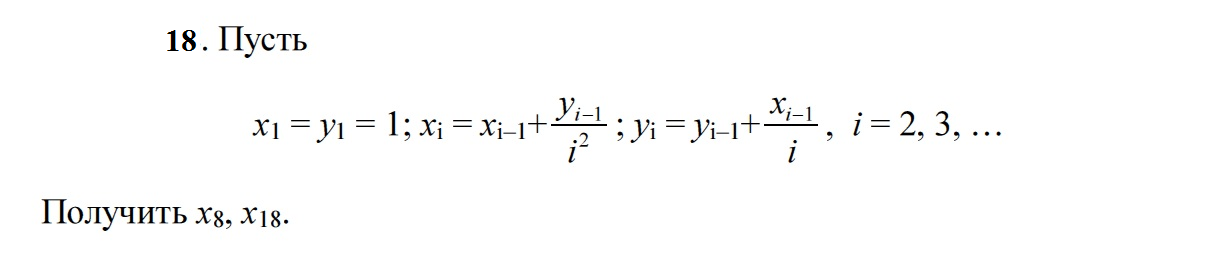


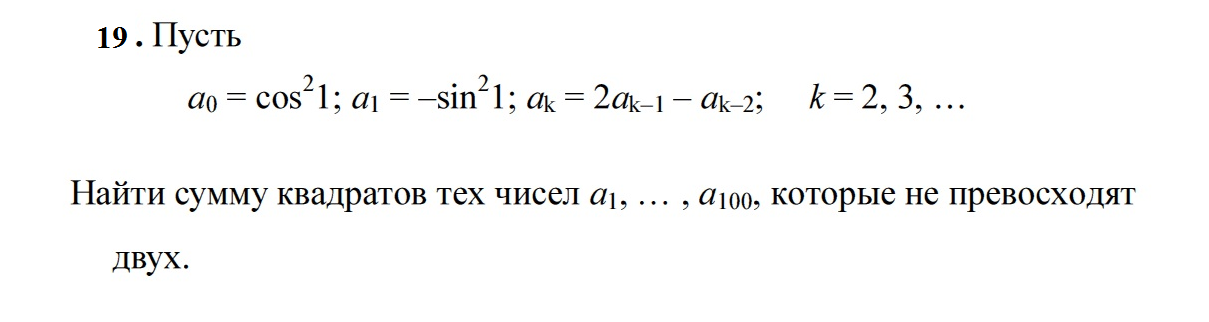


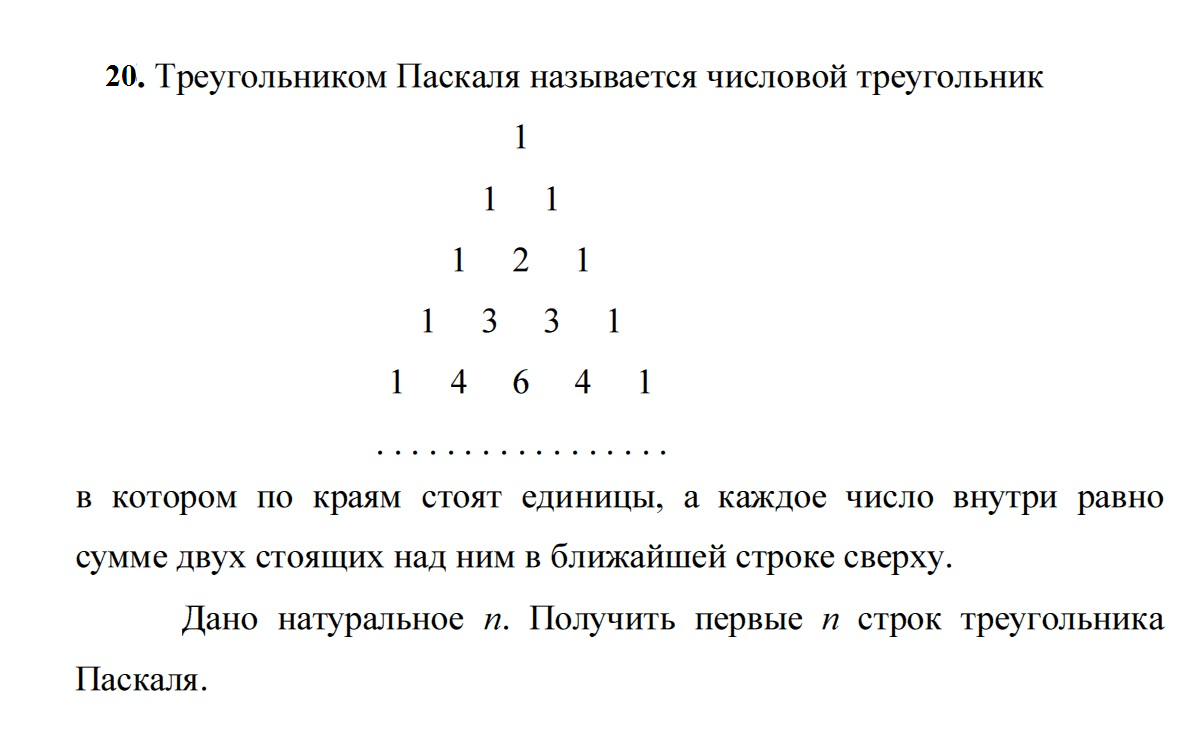












**21.** Реализовать рекурсивную функцию, которая определяет количество вхождений заданного символа *ch*в строку *str*. Функция получает 2 параметра:

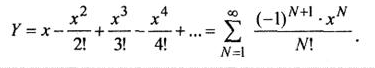
* символ *ch*, который нужно сравнить с каждым символом строки*str*;
* строка *str*.

Переменная *num* - счетчик символов.

**22.** Разработать рекурсивную функцию перевода числа из системы исчисления с основой **m** в систему исчисления с основой **n**. Функция должна получать следующие параметры:

* число ***num***, которое нужно перевести из системы исчисления **m** в систему ***n***;
* целочисленное беззнаковое число ***m***, задающее номинал системы исчисления с основой ***m***. Причем 2<***m***≤9;
* целочисленное беззнаковое число ***n*,** задающее номинал системы исчисления с основой ***n***. Причем 2<***n***≤9.

**23.** Дано действительное число x, ε>0. Вычислить сумму членов ряда c заданной точностью ε.



**24.** Разработать рекурсивную функцию, которая определяет, является ли заданное натуральное число **N** простым. Простое число, это число **N**>1 которое не имеет других делителей кроме 1 и самого себя.

**25.** Дано натуральное число ***n*** и действительное число ***х***. Написать рекурсивную процедуру для вычисления

**1-x(1/2)+x2(1\*3)/(2\*4)-x3(1\*3\*5)/(2\*4\*6)+…**

**+****(-1)nxn(1\*3\*…\*2n-1)/(2\*4\*…\*2n) .**

**26.** Дано натуральное число ***n*** и действительное число ***х***. Написать рекурсивную процедуру для вычисления

**1 + x2/2! + x4/4! +…+ x2n/(2n)!**

**27.** Дано натуральное число ***n*** и действительное число ***х***. Написать рекурсивную процедуру для вычисления

**(x-1)/x+(x-1)2/2x2+(x-1)3/3x3+…+(x-1)n/nxn**

**28.** Дано натуральное число ***n*** и действительное число ***х***. Написать рекурсивную процедуру для вычисления

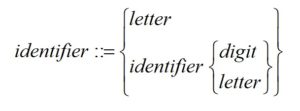
**1 – x2/ 2! + x4/4! + …+ (-1)nx2n/(2n)!**

**29.** Дано натуральное число ***n*** и действительное число ***х***. Написать рекурсивную процедуру для вычисления суммы слагаемых.

Выражение для ***n*** -го слагаемого имеет вид **Sn=(-1)n+1x2n-1/(2n+1)!,** ряд начинается с ***n* =0,** **S0=x.**

**30.** Разработать рекурсивную функцию, которая реализует синтаксический анализатор для понятия идентификатор:

* первый символ идентификатора обязательно буква латинского алфавита;
* второй, третий и т. д. символ идентификатора может быть буквой или цифрой. Общая формула идентификатора, следующая:

[](https://www.bestprog.net/wp-content/uploads/2019/01/recursion_04_.jpg)

где

* *identifier* – некоторый идентификатор;
* *letter* – некоторая буква латинского алфавита (от ‘a’ до ‘z’ или от ‘A’ до ‘Z’);
* *digit* – цифра от ‘0’ до ‘9’.