**3.1 Задания на вычисление суммы сходящегося ряда**

В заданиях (табл. 1) ряд может быть представлен как последова- тельностью членов ряда, так и в общем виде 



Примечание: *x*, *p* – это константы, объявленные в программе; *n* –

натуральные числа, принимающие значения в интервале [1; ∞].

# 3.2 Задания на нахождение корней уравнений

В заданиях (табл. 2) используются следующие сокращения:

*A* – метод половинного деления;



* 1. **Задания на одномерные массивы**

В одномерном массиве заменить отрицательные элементы на стоящие перед ними. Если первый элемент отрицательный, заменить на 0.

# Задания на двумерные массивы

Дан массив размера n × n. Значение элементов массива запол- няются случайными числами. Определить сумму минимальных и максимальных элементов главной и побочной диагонали.

5.1 **Необходимо создать метод, который:** определяет, сколько раз встречается заданная буква в слове, и выводит порядковые номера буквы в слове;

**6.1 ПОСТРОЕНИЕ РЕКУРСИВНЫХ АЛГОРИТМОВ**

Дано дерево глубины *n*, каждая вершина (кроме висячих) которого имеет *k* (не более 10) непосредственных потомков (нумеруют от 1 до *k*). Корень дерева имеет номер 0. Необходимо вывести на экран все возможные пути, ведущие от корня к листьям. Перебирать пути, начиная с «самого левого» и заканчивая «самым правым» (при этом первыми заменять конечные элементы пути).

**7.1 МЕТОДЫ СОРТИРОВКИ ДАННЫХ**

*Методы сортировки:* 1 – обменом;

*Содержимое массива:* A – прописные латинские;

*Направление сортировки:* ¯ - по убыванию.

**8.1 РАБОТА СО СТРОКАМИ**

Дана строка. Вывести подстроку, расположенную между первым и вторым пробелом исходной строки. Если строка содержит только один пробел, то вывести пустую строку.