

БОЛЬШОЕ ДОМАШНЕЕ ЗАДАНИЕ ПО ПРОГРАММИРОВАНИЮ НА C++

Решения присылайте по адресу shad.cplusplus@gmail.com. Дедлайн для студентов из Москвы: 18 февраля. Дедлайн для студентов из Киева, Минска, Харькова и заочников: 25 февраля. Оценка за задание будет поставлена во 2-й семестр.

Для определения номера своей задачи воспользуйтесь уже знакомой вам из курса «Алгоритмы и структуры данных поиска» функцией:

```
int getProblemNumber(const string& text, int numberOfProblems) {
    int result = 0;
    for (int i = 0; i < text.length(); ++i) {
        result = result * 239 + text[i];
    }
    return (result % numberOfProblems + numberOfProblems) %
        numberOfProblems + 1;
}
```

Первый параметр — ваши фамилия и имя, записанные латиницей. Например: `Zobnin Alexey`. Второй параметр — 4 (количество задач). Укажите свой первый параметр в письме с решением.

Задача 1. Создайте класс «Длинное целое число» для хранения целых чисел любого размера в заданной системе счисления. Основание системы счисления сделайте шаблонным параметром. Перегрузите арифметические операторы для сложения, вычитания, умножения и деления с остатком. Добавьте методы (или перегрузите операторы) для вычисления факториала и целой степени и для вычисления НОД двух таких чисел. Добавьте возможность преобразования чисел из одной системы счисления в другую. Реализуйте константный и неконстантный итераторы для перебора цифр этого числа. Не забудьте про оператор `<<` для вывода числа в поток. В качестве приложения реализуйте на основе этого класса простейший стековый калькулятор.

Задача 2. Создайте класс «Прямоугольная матрица» с разреженным хранением данных (то есть, храните только ненулевые элементы строк матрицы). Тип матричных элементов и размеры матрицы сделайте шаблонными. Класс должен поддерживать арифметические операции над матрицами (сложение, вычитание, умножение матриц, умножение на коэффициент), сравнение матриц, перегруженные операторы для получения матричного элемента в заданной позиции, приведение к ступенчатому виду, потоковый ввод и вывод. Специализация этого класса для квадратных матриц должна поддерживать вычисление определителя и обращение. Создайте конструкторы для создания случайной матрицы с заданной долей разреженности (то есть, с заданной долей ненулевых элементов). Реализуйте константный и неконстантный итераторы для перебора ненулевых матричных элементов. В качестве приложения класса реализуйте функцию, которая по матрице коэффициентов и столбцу свободных членов находит решение соответствующей системы линейных уравнений (быть может, зависящее от свободных параметров) или сообщает, что решения нет. Придумайте собственный формат хранения решения для этого метода.

Задача 3. Создайте класс «Многочлен от нескольких переменных» с разреженным хранением данных (то есть, храните только ненулевые слагаемые). Тип коэффициентов и количество переменных сделайте шаблонными. Класс должен поддерживать арифметические операции (сложение, вычитание, умножение, деление на коэффициент), вычисление полной степени, вычисление частных производных по переменным, вычисление полиномиального коэффициента перед данной степенью указанной переменной, получение коэффициента перед заданным мономом, а также иметь перегруженный оператор `<<` для вывода в поток. Реализуйте константный и неконстантный итераторы для перебора ненулевых слагаемых многочлена. Создайте специализацию этого класса для многочленов от одной переменной и реализуйте для нее алгоритм вычисления неполного частного и остатка от деления многочленов, а также алгоритм для вычисления наибольшего общего делителя. В качестве приложений напишите функцию, находящую все целые корни многочлена от одной переменной, а также функцию, проверяющую, имеет ли многочлен от одной переменной кратные корни.

Задача 4. Создайте класс «Дерево» на основе классов стандартной библиотеки. (У каждого узла дерева может быть произвольное количество дочерних узлов.) Дерево должно быть контейнером: в его узлах должны храниться элементы. Тип этих элементов сделайте шаблонным. Введите операции для получения, добавления и удаления узла, для получения его предка и потомков. Класс должен позволять вычислять высоту дерева, извлекать поддеревья. Переопределите оператор << для печати дерева в выходной поток. Реализуйте константный и неконстантный итераторы для навигации по дереву в разных направлениях. В качестве приложения напишите «имитатор командной строки», знающий команды `dir`, `cd`, `md`, `rd` для просмотра, изменения, создания и переименования (удаления) «каталогов» некоторой «файловой системы», хранящейся с помощью созданного класса.