

1.

Определить класс CIntN для работы с целыми беззнаковыми числами, состоящими из N десятичных цифр.

Внутри класса число должно быть реализовано с помощью указателя на тип char. Число N и размер отведенной памяти задаются в конструкторе класса.

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), сложения, вычитания, «, инкремент ++ и декремент - - (справа и слева), (умножение и деление на 10).

При сложении количество значащих десятичных цифр результата (N) может отличаться от N аргументов. Вычитание свести к сложению через дополнительный код.

В операторе вычитания использовать gvalue ссылку.

2. Определить класс CMatrix для работы с квадратной матрицей над полем Z_2 . Матрица будет определять множество решений соответствующей ее строкам системы линейных однородных уравнений. Размер матрицы и размер отведенной памяти задаются в конструкторе класса. Внутри класса матрица должна быть реализована с помощью указателя (int **).

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент ++ и декремент - - (справа и слева), увеличивающие (путем добавления нулевой строки и столбца) и уменьшающие (удаление последних строки и столбца, если это возможно) размер матрицы на 1, сложения, определяющий матрицу, для которой множество решений является пересечением решений систем линейных уравнений слагаемых, «.

При сложении матриц разного размера матрицу большего размера (после приведения к ступеньчатому виду) обрезать до размера меньшей матрицы (проекция линейных пространств).

В операторе сложения использовать gvalue ссылку.

В отдельном файле должен быть написан тест на данный класс.

3. Определить класс CText для работы с массивом строк (текстом). Количество строк текста и размер отведенной памяти под (каждую) строку задаются в конструкторе класса.

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент ++ и декремент - - (справа и слева, дописывание в конец всех строк текста символ ! и удаление последнего символа в каждой строке, если это возможно), сложения (конкатенация, или склейка строк текста).

В операторе декремент использовать gvalue ссылку.

В отдельном файле должен быть написан тест на данный класс.

4.

Определить класс CPoly для работы с многочленом от одной переменной с коэффициентами из поля вычетов Z_p , p – простое число, которое задается с помощью оператора #define. Степень многочлена и размер отведенной памяти задаются в конструкторе класса. Внутри класса полином должен быть реализована с помощью указателя (int *).

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, унарный плюс (возвращает профифференцированный полином), инкремент ++ и декремент - - (справа и слева), (результат – проинтегрированный и продифференцированный многочлен), сложения и вычитания. Если в операторе ++ требуется обратить 0 в поле Z_p , выбросить исключение.

В операторе унарный плюс использовать gvalue ссылку.

В отдельном файле должен быть написан тест на данный класс.

5. Определить класс CString для работы со строкой. Длина строки и размер отведенной памяти задаются в конструкторе класса. Внутри класса строка должна быть реализована с помощью указателя (char *).

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент ++ и декремент - - (справа и слева), (добавление символа ! в конец строки и обрезание последнего символа, если это возможно), сложения (конкатенация), умножения (слева и справа) строки на беззнаковое целое число (оно равносильно сложению строки с собой нужное число раз), унарный минус (возвращает строку, в которой удален первый символ).

В операторе унарный минус использовать gvalue ссылку.

В отдельном файле должен быть написан тест на данный класс.

6. Определить класс CMatrix для работы с квадратной матрицей целых чисел. Внутри класса матрица должна быть реализована с помощью указателя (int **). Размер матрицы и размер отведенной памяти задаются в конструкторе класса.

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент ++ и декремент - - (справа и слева), увеличивающие и уменьшающие (если это возможно) размер матрицы на 1 путем добавления в качестве последних рандомной строки и столбца (удаляющие последние столбец и строку), сложения, вычитания матриц, умножения матрицы (слева и справа) на число.

При сложении и вычитании размер результата – это минимум из размеров исходных матриц, лишние строки матрицы большего размера игнорируются.

В операторе вычитания использовать gvalue ссылку.

В отдельном файле должен быть написан тест на данный класс.

7. Определить класс CPoly2 для работы с полиномом с целыми коэффициентами от 2 переменных степени не выше N, N и размеротведенной памяти задаются в конструкторе класса. Внутри класса полином должен быть реализован с помощью указателя (int **).

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент ++ и декремент - - (справа и слева), (дифференцирование по переменным x и y). операторы сложения, вычитания.

В операторе сложения использовать gvalue ссылку.

В отдельном файле должен быть написан тест на данный класс с распечаткой полинома по степеням переменных.

8. Определить классы CVect для работы с массивом векторов с вещественными координатами на плоскости и CAngl для работы с массивом углов между векторами. Длины массивов и размер отведенной памяти задаются в конструкторах классов.

В классах должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент ++ и декремент - - (справа и слева), увеличивающие и уменьшающие (если это возможно) длину массива путем дублирования (вычеркивания) последнего элемента, операторы сложения (CVect и CAngl, возвращающий CVect, а также CAngl и CVect, возвращающий CVect), вычитания (CVect из CVect, возвращающий CAngl).

При сложении и вычитании длина результата – это минимум из длин исходных массивов.

В одном из операторов сложения использовать gvalue ссылку.

В отдельном файле должен быть написан тест на данные классы.

9. Определить класс `CArr` для работы с вектором (массивом) целых чисел. Длина массива и размер отведенной памяти задаются в конструкторе класса. Внутри класса массив должен быть реализован с помощью указателя (`int *`).

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент `++` и декремент `--` (справа и слева), дублирующие (удаляющие) последний элемент массива, если он существует, вычитания (массив, полученный удалением из первого массива всех элементов второго с учетом повторов).

В операторе вычитания использовать `rvalue` ссылку.

В отдельном файле должен быть написан тест на данный класс.

10. Определить классы `CPoint` для работы с массивом целочисленных точек на плоскости и `CDist` для работы с вектором расстояний между целочисленными точками.

Внутри классов массивы должны быть реализованы с помощью указателя на классы "точка плоскости с целочисленными координатами" и "2-мерное расстояние". Длина массива и вектора и размер отведенной памяти задаются в конструкторе класса.

В классах должны быть определены необходимые конструкторы (в том числе конструкторы копирования и перемещения), деструкторы, операторы присваивания (копированием и перемещением), сложения (`CPoint` и `CDist`, возвращающий `CPoint`, а также `CDist` и `CPoint`, возвращающий `CPoint`), вычитания (`CPoint` из `CPoint`, возвращающий `CDist`), «, инкремент `++` и декремент `--` (справа и слева), увеличивающие и уменьшающие (если это возможно) длину вектора путем дублирования последнего (удаления последнего) элемента.

В одном из операторов сложения использовать `rvalue` ссылку.

При сложении и вычитании длина результата – это минимум из длин аргументов.

В отдельном файле должен быть написан тест на данные классы.

11. Определить класс CRat для работы с вектором (массивом) несократимых дробей вида p_i/q_i , где p_i — целое, q_i — натуральное.

Внутри класса вектор должен быть реализован с помощью указателя на предварительно реализованный класс "несократимая дробь". Длина вектора и размер отведенной памяти задаются в конструкторе класса.

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), сложения, вычитания, «, инкремент ++ и декремент - - (справа и слева), увеличивающие и уменьшающие (если это возможно) длину вектора путем добавления в конец суммы его элементов (удаления последнего элемента).

При сложении и вычитании длина результата — это минимум из длин исходных векторов.

В операторе сложения использовать gvalue ссылку.

В отдельном файле должен быть написан тест на данный класс.

12.

Определить класс CArr для работы с вектором (массивом) целых чисел. Внутри класса массив должен быть реализован с помощью указателя. Длина вектора и размер отведенной памяти задаются в конструкторе класса.

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент ++ и декремент - - (справа и слева), увеличивающие и уменьшающие (если это возможно) длину вектора путем добавления в начало вектора числа 0 (удаления первого числа), сложения вектора с целым числом (добавление числа в конец массива), унарного минуса - (удаление последнего элемента, если это возможно).

В операторе унарный минус использовать gvalue ссылку.

В отдельном файле должен быть написан тест на данный класс.

13.

Определить класс CSet для работы с битовым множеством целых чисел в диапазоне от 0 до N. Внутри класса множество должно быть реализовано с помощью указателя (unsigned int *) или (unsigned long long *). Число N и размер отведенной памяти задается в конструкторе класса. Принадлежность числа множеству означает, что бит, соответствующий этому числу, равен 1, в случае нулевого бита число не принадлежит множеству.

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент ++ и декремент - - (справа и слева), увеличивающие и уменьшающие (если это возможно) верхнюю границу диапазона, сложения (объединение множеств), вычитания (пересечения).

При сложении верхняя граница диапазона результата равна максимуму из соответствующих границ слагаемых, при вычитании – минимуму.

В операторе вычитания использовать gvalue ссылку.

В отдельном файле должен быть написан тест на данный класс.

14. Определить класс CArr для работы с упорядоченным по возрастанию массивом целых чисел. Длина массива и размер отведенной памяти задаются в конструкторе. В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент ++ и декремент - - (справа и слева – дублирование последнего и удаление первого элемента, если это возможно), «, оператор сложения (массив, полученный слиянием слагаемых).

В операторе декремент использовать gvalue ссылку.

В отдельном файле должен быть написан тест на данный класс.

15. Определить класс $CInt_p$ для работы с вектором чисел в поле вычетов по модулю p , (Z_p , p – простое число, которое определяется через `#define`). Внутри класса вектор должен быть реализован с помощью указателя. Длина вектора и размер отведенной памяти задаются в конструкторе класса.

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), сложения, вычитания, умножения и деления (деление не должно быть основано на переборе), `«`, инкремент `++` и декремент `--` (справа и слева), дублирующие и удаляющие (если это возможно) первый элемент вектора. В ситуации деления на ноль нужно выбросить исключение.

При сложении, вычитании, умножении и делении длина результата – это минимум из длин исходных векторов.

В операторе вычитания использовать `gvalue` ссылку.

В отдельном файле должен быть написан тест на данный класс.

16. Определить класс $CStr$ для работы со строкой. Размер отведенной памяти задается в конструкторе класса.

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), `«`, инкремент `++` и декремент `--` (справа и слева), дублирующие (удаляющие) последний элемент строки, если он существует, операторы сложения (слева и справа) строки с символом (добавление символа в начало/конец строки), вычитания из строки символа (удаление всех его вхождений), `«`.

В операторе вычитания использовать `gvalue` ссылку.

В отдельном файле должен быть написан тест на данный класс.

17. Определить класс `CComplexMatrix` для работы с квадратной матрицей комплексных чисел. Размер матрицы и размер отведенной памяти задаются в конструкторе класса.

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент `++` и декремент `--` (справа и слева), увеличивающие и уменьшающие (если это возможно) размер матрицы на 1 путем добавления в качестве последних нулевых строки и столбца (удаляющие последние столбец и строку), сложения, вычитания матриц, умножения матрицы (слева и справа) на комплексное число.

При сложении и вычитании размер результата – это минимум из размеров исходных матриц, лишние строки матрицы большего размера игнорируются.

В операторе вычитания использовать `gvalue` ссылку.

В отдельном файле должен быть написан тест на данный класс.

18. Определить класс `CMatrix` для работы с криволинейной матрицей целых чисел. Количество строк матрицы задаются в конструкторе класса.

В классе должны быть определены необходимые конструкторы (в том числе конструктор копирования и перемещения), деструктор, операторы присваивания (копированием и перемещением), «, инкремент `++` и декремент `--` (справа и слева, добавление в конец всех строк матрицы числа 0 и удаление последнего числа в каждой строке, если это возможно, вычитания (матрица из строк, которые есть в обоих аргументах).

В операторе вычитания использовать `gvalue` ссылку.

В отдельном файле должен быть написан тест на данный класс.