

Задание 6

Спроектировать 2-мерную разреженную бесконечную матрицу, заполненную значениями по умолчанию. Матрица должна хранить только занятые элементы - значения которых хотя бы раз присваивались. Присвоение в ячейку значения по умолчанию освобождает ячейку.

Необходимо уметь отвечать на вопрос - сколько ячеек реально занято?

Необходимо уметь проходить по всем занятым ячейкам. Порядок не имеет значения. Возвращается позиция ячейки и ее значение.

При чтении элемента из свободной ячейки возвращать значение по умолчанию.

Пример:

```
// бесконечная матрица int заполнена значениями -1
Matrix<int, -1> matrix;
assert(matrix.size() == 0); // все ячейки свободны

auto a = matrix[0][0];
assert(a == -1);
assert(matrix.size() == 0);

matrix[100][100] = 314;
assert(matrix[100][100] == 314);
assert(matrix.size() == 1);

// выведется одна строка
// 100100314
for(auto c: matrix)
{
    int x;
    int y;
    int v;
    std::tie(x, y, v) = c;
    std::cout << x << y << v << std::endl;
}
```

При запуске программы необходимо создать матрицу с пустым значением 0, заполнить главную диагональ матрицы (от [0,0] до [9,9]) значениями от 0 до 9.

Второстепенную диагональ (от [0,9] до [9,0]) значениями от 9 до 0.

Необходимо вывести фрагмент матрицы от [1,1] до [8,8]. Между столбцами пробел. Каждая строка матрицы на новой строке консоли.

Вывести количество занятых ячеек.

Вывести все занятые ячейки вместе со своими позициями.

Опционально реализовать N-мерную матрицу.

Опционально реализовать каноническую форму оператора `=`, допускающую выражения `((matrix[100][100] = 314) = 0) = 217`

Самоконтроль

- индексация оператором `[]`

- количество занятых ячеек должно быть 18 после выполнения примера выше

Проверка

Задание считается выполненным успешно, если после анализа кода, установки пакета и запуска приложения появился фрагмент матрицы, количество ячеек и список всех значений с позициями.