

```
1. Реализуйте АТД Стек на односвязном списке (forward_list).
template <typename T>
class Stack {
private:
    std::forward_list<T> st;
public:
    Stack(){};
    // Проверяет, пуст ли стек.
    bool isEmpty() const {
        return st.empty();
    }
    // Добавляет элемент на вершину стека.
    void push(const T& element) {
        st.push_front(element);
    }
    // Возвращает элемент на вершине стека.
    T front() const {
        if (isEmpty())//Проверка на пустоту стека.
            throw std::out_of_range("Stack is empty.");
        return st.front();
    }
    // Удаляет элемент с вершины стека
    void pop() {
        if (isEmpty()) //Проверка на пустоту стека.
            throw std::out_of_range("Stack is empty.");
        st.pop_front();
    }
};
```

12. Определить АД Матрица, обеспечивающий метод $[i, j]$ для доступа к элементам матрицы. В конструкторе задаются размеры матрицы.

Реализовать матрицу через vector размером $N \cdot M$. Определить операцию $+$. Сравнить время сложения матриц размером 1000×1000 , меняя порядок циклов (строки/столбцы и столбцы/строки) для уровня оптимизации ОЗ. Результаты записать в таблицу, в которой будет указан порядок выполнения циклов и время выполнения в мкс.

```
class Matrix
{
    int n, m;
    vector<double> v;

public:
    Matrix(int n, int m) : n(n), m(m)
    {
        v.resize(n * m);
    }
    double &operator[] (pair<int, int> index)
    {
        auto [i, j] = index;
        if (i < 0 || i > n || j < 0 || j > m)
            throw runtime_error("Invalid Index");

        return v[i * n + j];
    }
    Matrix operator+(const Matrix &other)
    {
        if (n != other.n || m != other.m)
            throw runtime_error("Invalid Sizes");

        Matrix answer(n, m);
        for (int i = 0; i < n; i++)
            for (int j = 0; j < m; j++)
                answer[{i, j}] = v[i * n + j] + other.v[i * n + j];
        return answer;
    }
};
```

| | |
|-------------------|-------------------|
| for(i){ for(j) } | for(j){ for(i) } |
| 36760 микросекунд | 38213 микросекунд |

(Средние за 100 попыток)