### Тренировочное задание по программированию: StackVector

### Условие

В блоке про модель памяти мы говорили, что динамическая память нужна для хранения тех объектов, которые должны жить дольше, чем создавшая их функция. Кроме того мы говорили, что можно создать такой вектор, который не использует память из кучи, а хранит все свои объекты на стеке. В этой задаче вам предстоит это сделать. Вы совсем недавно познакомились с контейнером array, который хранит все свои данные на стеке. Реализуйте интерфейс вектора поверх него. Более формально, реализуйте шаблон класса template <typename T, size\_t N> StackVector<T, N>, где T — тип элемента вектора, а N — его максимальная вместимость. Класс StackVector должен иметь следующий интерфейс:

template <typename T, size\_t N>

class StackVector {

public:

  explicit StackVector(size\_t a\_size = 0);

  T& operator[](size\_t index);

  const T& operator[](size\_t index) const;

  ??? begin();

  ??? end();

  ??? begin() const;

  ??? end() const;

  size\_t Size() const;

  size\_t Capacity() const;

  void PushBack(const T& value);

  T PopBack();

};

* Метод Capacity должен возвращать вместимость вектора, то есть количество объектов, которое в него в принципе может поместиться.
* Метод Size должен возвращать текущее количество объектов в векторе
* Конструктор принимает размер вектора (по аналогии со стандартным вектором). Если аргумент конструктора оказался больше вместимости вектора, конструктор должен выбрасывать исключение [invalid\_argument](https://en.cppreference.com/w/cpp/error/invalid_argument)
* Методы begin/end должны возвращать итераторы на начало и текущий конец вектора; их тип не указан, вы должны выбрать его самостоятельно
* Метод PushBack добавляет новый элемент в конец вектора. Если текущий размер вектора равен его вместимости, метод PushBack должен выбрасывать стандартное исключение [overflow\_error](https://en.cppreference.com/w/cpp/error/overflow_error)
* Метод PopBack уменьшает размер вектора на один и возвращает элемент вектора, который был последним. Если вектор пуст, метод PopBack должен выбрасывать стандартное исключение [underflow\_error](https://en.cppreference.com/w/cpp/error/underflow_error)

Интерфейс класса StackVector описан в файле stack\_vector.h, приложенном к этой задаче. Часть требований к нему сформулирована в виде юнит-тестов в файле stack\_vector.cpp.

Обратите внимание, как отличается поведение метода PushBack в классе StackVector от метода PushBack в классе SimpleVector, который вы реализовывали ранее в блоке про модель памяти. SimpleVector::PushBack выделял дополнительную память в куче, если размер вектора оказывался равен ёмкости. В случае с классом StackVector мы должны на этапе компиляции задать максимальную ёмкость вектора и, если она оказывается исчерпана, нам больше неоткуда взять память, и мы просто бросаем исключение. В этом состоит недостаток реализации вектора на стеке по сравнению с обычным вектором.

С другой стороны, файл stack\_vector.cpp содержит бенчмарк, который демонстрирует преимущество StackVector перед обычным вектором. Этот бенчмарк моделирует ситуацию, когда мы считываем из потока количество объектов N, а затем добавляем в вектор N объектов с помощью метода push\_back (тестовые данные разложены в vector<vector<int>>, чтобы исключить из бенчмарка время, необходимое на парсинг). Когда вы реализуете шаблон StackVector, сравните его производительность с вектором.

На проверку пришлите заголовочный файл с реализацией шаблона StackVector.

### Заготовка решения

[stack\_vector.h](https://d3c33hcgiwev3.cloudfront.net/Tx4fD2TMEei5FgrpHNEYyg_4f49e93064cc11e89930437d486edb8c_stack_vector.h?Expires=1617062400&Signature=iOL1l4FSssQFJYF3eqgl4Y5hKY5Gr~GJx5-wrIpva7r3H~09mxIehOd-jl-GnrFxu5g1YWxxr7pqtVu2yDOQ5xOIT4~QLjfV7d82p5RbJUMjNQqgAsqzlxNviaTc4eQWoMfCgvpk~6-vqucxZoqoEoZNR2f5Q7atm5QBA6853cA_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

[stack\_vector.cpp](https://d3c33hcgiwev3.cloudfront.net/ubWLDnqyEeiC-BK3XY5K_A_ba192ea07ab211e8a9f45d50491009f0_stack_vector.cpp?Expires=1617062400&Signature=J4pi8Wt~AN-QkczaP~bKLRxDxCp9vEeHKoqXcmAOYhpEqXw-yGyhC3G9DEcZTVuv1VqaWpjNSyFiGvITFDwENtDbrERH3YY2Uz4LopEP07oBsYSoXSf6JFDNhEKREl928aVrdCKvHBrOV3SjzkRmK7jlkH5apfFLy52sn1tG1v8_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

### Замечание

Заголовочный файл, который вы пришлёте на проверку, не должен подключать файлы <vector>, <list>, <forward\_list>, <deque>, <map>. Если у вас будет подключён один из этих файлов, вы получите ошибку компиляции.