**Принципы работы контейнера std::vector из библиотеки <vector>**

**Представление в памяти.** Вектор — это динамический массив, который может изменять свой размер при добавлении или удалении дополнительных элементов. У вектора есть метод .size() – количество реальных объектов (размер), и .capacity() – количество объектов, под которые зарегистрирована память (ёмкость). Размер всегда больше или равен ёмкости, так как вектор резервирует память с запасом.

\*Схема представления вектора в памяти\*

**Вставка.** При добавлении элемента (.push\_back(), .insert() и др.) если размер меньше ёмкости, он помещается в свободную память. Если размер равен ёмкости, то происходит перераспределение. Используется политика экспоненциального роста, то есть ёмкость нового блока памяти увеличивается в 2 или 1,5 раза (зависит от компилятора). Старые элементы копируются в новый блок.

**Удаление.** Удаление элемента (.erase(), .pop\_back(0)) не вызывает перераспределение. Удаленный объект будет уничтожен, но память останется принадлежать вектору. Метод .shrink\_to\_fit() позволяет освободить лишнюю память.



Рис.1. Запуск тестовой программы с выводом адресов

**Приложение А: проведение ксперимента**

