Задача 1-1 (20 баллов). Предложите реализацию *стека* на основе (одного) массива, которая поддерживает операции добавления в конец и удаления из конца. Требуется, чтобы *емкость* (количество выделенных ячеек памяти) стека в любой момент времени отличалась от фактического размера не более чем в константу раз, а учетная сложность операций добавления в конец и удаления из конца была константной.

Решение. Будет использоваться динамический массив с изменением размера (увеличением/уменьшением) размера в константу раз (напр., в 2 раза) при достижении `size`'а `capacity` или при уменьшении `size`'а в константу раз (минус единица) от `capacity`.

Это позволяет обеспечить, чтобы емкость стека была не более чем в константу раз больше его фактического размера и чтобы учетная сложность операций добавления и удаления была $O(1)$. При этом, стоимость операций textttpush и `pop` имеют амортизированную сложность $O(1)$.

Для доказательства амортизированной стоимости операций `push` и `pop` в стеке с динамическим изменением размера используем **метод банкира** (accounting method). Каждой операции присваивается амортизированная стоимость 3 единицы, что покрывает её фактическую стоимость и накапливает "кредит"для более затратных операций изменения размера.

Если при добавлении (`push`) массив не заполнен, фактическая стоимость составляет 1, а оставшиеся 2 единицы идут в "кредит". Если массив заполнен, емкость удваивается и элементы копируются, что стоит n операций (где n — текущий размер). Однако накопленные $2n$ единиц кредита с предыдущих `push` покрывают затраты на удвоение.

Аналогично, при удалении (`pop`) амортизированная стоимость 3 покрывает 1 единицу на удаление и оставляет 2 в "кредит". Если размер уменьшается до capacity/4, емкость делится пополам и требуется n операций для копирования, что компенсируется накопленным кредитом $2n$.

Таким образом, амортизированная стоимость операций `push` и `pop` — 3, обеспечивая амортизированную сложность $O(1)$.