

# ВШЭ АиСД 2021. Стек, очередь (noSTL)

4 дек 2021, 18:58:06

старт: 30 сен 2021, 11:30:00

финиш: 10 окт 2021, 23:59:59

длительность: 10д. 12ч.

начало: 30 сен 2021, 11:30:00

конец: 10 окт 2021, 23:59:59

## А. Безошибочный стек (0.2)

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Реализуйте структуру данных (класс) "стек" на основе целочисленного динамического массива.

Требуется разработать программу, содержащую описание стека и моделирующую работу стека в соответствии с указанными ниже командами.

Программа считывает последовательность команд и, в зависимости от команды, выполняет ту или иную операцию.

После выполнения каждой команды программа должна вывести одну строку. Набор команд следующий:

1. **push n**: Добавить в стек число *n* (значение *n* задается через пробел). Программа должна вывести **ok**.
2. **pop**: Удалить из стека головной элемент. Программа должна вывести его значение.
3. **back**: Программа должна вывести значение головного элемента стека, не удаляя его из стека.
4. **size**: Программа должна вывести количество элементов в стеке.
5. **clear**: Программа должна очистить стек и вывести **ok**.
6. **exit**: Программа должна вывести **bye** и завершить работу.

Перед исполнением операций **back** и **pop** программа должна проверять, содержится ли в стеке хотя бы один элемент.

Если во входных данных встречается операция **back** или **pop**, и при этом стек пуст, то программа должна вместо числового значения вывести строку **error**.

Общее число элементов в стеке не превышает 10000.

## Формат ввода

Вводятся команды управления реализованным вами стеком, построчно.

## Формат вывода

Программа должна вывести протокол работы стека, по одному сообщению в строке.

## Пример

Ввод Вывод 

push 1

ok

back

1

exit

bye

```
1 #include <iostream>
2 #include <string>
3
4 class Stack {
5 private:
6     int64_t *stack;
7     int64_t size;
8     // int64_t top;
9
10 public:
11     explicit Stack(int64_t sizeStack) {
12         stack = new int64_t[sizeStack];
13         // top = 0;
14         size = 0;
15     }
16
17     ~Stack() = default;
18     void Push(const std::string &data) {
19         stack[size] = std::stoi(data.substr(5, data.size()));
20         ++size;
21
22         std::cout << "ok" << '\n';
23     }
24
25     void Pop() {
26         if (size == 0) {
27             std::cout << "error" << '\n';
28             return;
29         }
30
31         std::cout << stack[size - 1] << '\n';
32
33         stack[size - 1] = 0;
34         --size;
35     }
36
37     void Back() {
38         if (size == 0) {
```