

# Яндекс.Практикум, Алгоритмы: Спринт 1.

## Базовые структуры данных

🕒 5 авг 2020, 00:37:00  
старт: 1 авг 2020, 18:02:26

### Q. Дек

	Все языки	Oracle Java 8
Ограничение времени	0.1 секунда	0.2 секунды
Ограничение памяти	39Mb	64Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

Гоша решил реализовать структуру данных Дек, максимальный размер которого определяется заданным числом. Методы `push_back`, `push_front`, `pop_front`, `pop_back` работали корректно. Но, если в деке было много элементов, программа работала очень долго. Дело в том, что не все операции выполнялись за  $O(1)$ . Помогите Гоше! Напишите эффективную реализацию.

### Формат ввода

В первой строке записано количество команд  $n$  - целое число, не превосходящее 5000. Во второй строке записано число  $m$  - максимальный размер стека. Он не превосходит 1000. В следующих  $n$  строках записана одна из команд:

`push_back value`

`push_front value`

`pop_front`

`pop_back`

`value` - целое число, по модулю не превосходящее 1000.

### Формат вывода

При вызове команд `pop_front` и `pop_back` нужно вывести возвращаемое значение. Если они вызываются для пустого стека - напечатайте `'error'`. Если команда `push_back` или `push_front` вызывается для стека, размер которого равен максимально возможному, тоже нужно вывести `'error'`.

### Пример 1

Ввод	Вывод
4	861
4	-819
push_front 861	
push_front -819	
pop_back	
pop_back	

### Пример 2

Ввод	Вывод
------	-------


Ввод	Вывод
7	-855
10	720
push_front -855	844
push_front 720	
pop_back	
pop_back	
push_back 844	
pop_back	
push_back 823	

### Пример 3

Ввод	Вывод
6	20
6	102
push_front -201	
push_back 959	
push_back 102	
push_front 20	
pop_front	
pop_back	

### Примечания

При реализации нельзя использовать связный список.

Язык GNU c++17 7.3 

Набрать здесь

Отправить файл

1

Отправить

Предыдущая