

## 18. Дек с защитой от ошибок

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Научитесь пользоваться стандартной структурой данных deque для целых чисел. Напишите программу, содержащую описание дека и моделирующую работу дека, реализовав все указанные здесь методы. Программа считывает последовательность команд и в зависимости от команды выполняет ту или иную операцию. После выполнения каждой команды программа должна вывести одну строчку. Возможные команды для программы:

push\_front n

Добавить (положить) в начало дека новый элемент. Программа должна вывести ok.

push\_back n

Добавить (положить) в конец дека новый элемент. Программа должна вывести ok.

pop\_front

Извлечь из дека первый элемент. Программа должна вывести его значение.

pop\_back

Извлечь из дека последний элемент. Программа должна вывести его значение.

front

Узнать значение первого элемента (не удаляя его). Программа должна вывести его значение.

back

Узнать значение последнего элемента (не удаляя его). Программа должна вывести его значение.

size

Вывести количество элементов в деке.

clear

Очистить дек (удалить из него все элементы) и вывести ok.

exit

Программа должна вывести bye и завершить работу.

Гарантируется, что количество элементов в деке в любой момент не превосходит 100. Перед исполнением операций pop\_front, pop\_back, front, back программа должна проверять, содержится ли в деке хотя бы один элемент. Если во входных данных встречается операция pop\_front, pop\_back, front, back, и при этом дек пуст, то программа должна вместо числового значения вывести строку error.

### Формат ввода

Вводятся команды управления деком, по одной на строке.

### Формат вывода

Требуется вывести протокол работы дека, по одному сообщению на строке

#### Пример 1

Ввод

Вывод

push\_back 1  
back  
exit

ok  
1  
bye

#### Пример 2

Ввод

Вывод

size  
push\_back 1  
size  
push\_back 2

0  
ok  
1  
ok

size	2
push_front 3	ok
size	3
exit	bye

## Пример 3

Ввод

Вывод

push_back 3	ok
push_front 14	ok
size	2
clear	ok
push_front 1	ok
back	1
push_back 2	ok
front	1
pop_back	2
size	1
pop_front	1
size	0
exit	bye

Язык GNU GCC 12.2 C++20

Набрать здесь

Отправить файл

```
1 #include <iostream>
2 #include <deque>
3
4 int main() {
5     std::string s;
6     std::deque<int> q;
7     while (std::cin >> s) {
8         if (int tmp; s == "push_front") {
9             std::cin >> tmp;
10            q.push_front(tmp);
11            std::cout << "ok\n";
12        } else if (int tmp; s == "push_back") {
13            std::cin >> tmp;
14            q.push_back(tmp);
15            std::cout << "ok\n";
16        } else if (int tmp; s == "pop_front") {
17            if (q.empty()) {
18                std::cout << "error\n";
19                continue;
20            }
21            tmp = q.front();
22            q.pop_front();
23            std::cout << tmp << "\n";
24        } else if (int tmp; s == "pop_back") {
25            if (q.empty()) {
26                std::cout << "error\n";
27                continue;
28            }
29            tmp = q.back();
30            q.pop_back();
31            std::cout << tmp << "\n";
32        } else if (s == "front") {
33            if (q.empty()) {
34                std::cout << "error\n";
35                continue;
36            }
37            std::cout << q.front() << "\n";
38        } else if (s == "back") {
```

Отправить

Предыдущая

Следующая