

# Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

🕒 4 апр 2024, 02:41:19  
старт: 28 окт 2023, 21:14:17  
финиш: 26 ноя 2023, 03:14:16  
длительность: 28д. 5ч.  
...

Объявления жюри

📘 Ваше участие в соревновании завершено. Вы можете дорешивать задачи и отправлять решения вне соревнования

Положение участников    Задачи    Посылки

## 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	ok
back	1
exit	bye

#### Пример 2

Ввод	Вывод
size	0
push_back 1	ok
size	1
push_back 2	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

- ✓

1. Гистограмма
- ✓

2. Красивая строка
- ✓

3. Коллекционер Диего
- ✓

4. Контрольная работа
- ✓

5. Хорошая строка
- ✓

6. Операционные системы lite
- ✓

7. SNTP
- ✓

8. Минимальный прямоугольник
- ✓

9. Сумма в прямоугольнике
- ✓

10. Скучная лекция
- ✓

11. Стек с защитой от ошибок
- ✓

12. Правильная скобочная последовательность
- ✓

13. Постфиксная запись
- ✓

14. Сортировка вагонов lite
- ✓

15. Великое Лайнландское переселение
- ✓

16. Очередь с защитой от ошибок
- ✓

17. Игра в пьяницу
- ✓

18. Дек с защитой от ошибок
- ✓

19. Хипуй
- ✓

20. Пирамидальная сортировка
- ✗

21. Три единицы подряд
22. Кузнечик
23. Калькулятор
24. Покупка билетов
25. Гвоздики
26. Самый дешевый путь
27. Вывести маршрут максимальной стоимости
28. Ход конём
29. Кафе
30. НОП с восстановлением ответа
- ✓

31. Поиск в глубину
- ✓

32. Компоненты связности
- ✓

33. Списывание
- ✓

34. Топологическая сортировка
- ✓

35. Поиск цикла
- ✓

36. Длина кратчайшего пути
- ✓

37. Путь в графе
- ✓

38. Блохи
- ✓

39. Путь спелеолога
- ✓

40. Метро

Язык Python 3.9 (PyPy 7.3.11) ▾

Набрать здесь    Отправить файл

```
1 class Stack():
2     def __init__(self):
3         self.head = None
4         self.tail = None
5         self.length = 0
6
7     def push(self, val):
8         if self.head is None:
9             self.head = [val, None, None]
10            self.tail = self.head
11        else:
12            self.tail[1] = [val, None, None]
13            self.tail[1][2], self.tail = self.tail, self.tail[1]
14            self.length += 1
15        return 'ok'
16
17    def pushleft(self, val):
18        if self.head is None:
19            self.head = [val, None, None]
20            self.tail = self.head
21        else:
22            self.head[2] = [val, None, None]
23            self.head[2][1], self.head = self.head, self.head[2]
24            self.length += 1
25        return 'ok'
26
27    def pop(self):
28        if self.tail is None: return 'error'
29        res = self.tail[0]
30
31        if self.tail[2] is not None:
32            self.tail[2][1], self.tail = None, None, self.tail[2]
33            self.length -= 1
34        else:
35            self.head = None
36            self.tail = None
37
38
```

Отправить    📘 осталось 100 попыток

Предыдущая

Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы
29 окт 2023, 14:43:27	95055892	18	Python 3.9 (PyPy 7.3.11)	OK	-	305ms	28.32Mb	-	-