

Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

19 фев 2023, 16:56:53

старт: 13 фев 2023, 15:00:00

финиш: 13 мар 2023, 17:59:59

до финиша: 22д. 1ч.

...

Объявления жюри

Положение участников Задачи Посылки

16. Очередь с защитой от ошибок

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

Научитесь пользоваться стандартной структурой данных queue для целых чисел. Напишите программу, содержащую описание очереди и моделирующую работу очереди, 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 строку.

Возможные команды для программы:

`push n`
Добавить в очередь число `n` (значение `n` задается после команды). Программа должна вывести `ok`.

`pop`
Удалить из очереди первый элемент. Программа должна вывести его значение.

`front`
Программа должна вывести значение первого элемента, не удаляя его из очереди.

`size`
Программа должна вывести количество элементов в очереди.

`clear`
Программа должна очистить очередь и вывести `ok`.

`exit`
Программа должна вывести `bye` и завершить работу.

Перед исполнением операций `front` и `pop` программа должна проверять, содержит ли в очереди хотя бы один элемент. Если во входных данных встречается операция `front` или `pop`, и при этом очередь пуста, то программа должна вместо числового значения вывести строку `error`.

Формат ввода

Вводятся команды управления очередью, по одной на строке

Формат вывода

Требуется вывести протокол работы очереди, по одному сообщению на строке

Пример 1

Ввод	Выход
<code>push 1</code>	<code>ok</code>
<code>front</code>	<code>1</code>
<code>exit</code>	<code>bye</code>

Пример 2

Ввод	Выход
<code>size</code>	<code>0</code>
<code>push 1</code>	<code>ok</code>
<code>size</code>	<code>1</code>
<code>push 2</code>	<code>ok</code>
<code>size</code>	<code>2</code>
<code>push 3</code>	<code>ok</code>
<code>size</code>	<code>3</code>
<code>exit</code>	<code>bye</code>

Пример 3

Ввод	Выход
<code>push 3</code>	<code>ok</code>

- 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

```
push 14          ok
size            2
clear           ok
push 1          ok
front           1
push 2          ok
front           1
pop              1
size            1
pop              2
size           0
exit            bye
```

Язык

Python 3.11.2

[Набрать здесь](#) [Отправить файл](#)

1

[Отправить](#)

осталось 100 попыток

[Предыдущая](#)

[Следующая](#)

Письлок нет

Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

19 фев 2023, 16:59:57

старт: 13 фев 2023, 15:00:00

финиш: 13 мар 2023, 17:59:59

до финиша: 22д.

Объявления жюри

...

Положение участников Задачи Посылки

17. Игра в пьяницу

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

В игре в пьяницу карточная колода раздается поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остается без карт – проигрывает. Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту ("шестерка берет туза"). Игрок, который забирает себе карты, сначала кладет под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды). Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует 10 карт, имеющих значения от 0 до 9, большая карта побеждает меньшую, карта со значением 0 побеждает карту 9.

Формат ввода

Программа получает на вход две строки: первая строка содержит 5 чисел, разделенных пробелами — номера карт первого игрока, вторая — аналогично 5 карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой.

Формат вывода

Программа должна определить, кто выигрывает при данной раздаче, и вывести слово first или second, после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении 10^6 ходов игра не заканчивается, программа должна вывести слово botva.

Пример 1

Ввод	Выход
1 3 5 7 9	
2 4 6 8 0	second 5

Пример 2

Ввод	Выход
2 4 6 8 0	
1 3 5 7 9	first 5

Пример 3

Ввод	Выход
1 7 3 9 4	
5 8 0 2 6	second 23

- 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

Язык Python 3.11.2

Набрать здесь Отправить файл

1

[Отправить](#)

 осталось 100 попыток

[Предыдущая](#)

[Следующая](#)

 Помощь нет

Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

19 фев 2023, 17:00:18
 старт: 13 фев 2023, 15:00:00
 финиш: 13 мар 2023, 17:59:59
 до финиша: 22д.
 ...

Положение участников Задачи Посылки

18. Дек с защитой от ошибок

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Выход	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 строку.

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, содержит 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.

- 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

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	Выход
<code>push_back 1</code>	<code>ok</code>
<code>back</code>	<code>1</code>
<code>exit</code>	<code>bye</code>

Пример 2

Ввод	Выход
<code>size</code>	<code>0</code>
<code>push_back 1</code>	<code>ok</code>
<code>size</code>	<code>1</code>
<code>push_back 2</code>	<code>ok</code>
<code>size</code>	<code>2</code>
<code>push_front 3</code>	<code>ok</code>
<code>size</code>	<code>3</code>
<code>exit</code>	<code>bye</code>

Пример 3

Ввод	Выход
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 ▼

1 осталось 100 попыток

1 Помощь нет

Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

19 фев 2023, 17:00:41

Объявления жюри

старт: 13 фев 2023, 15:00:00

финиш: 13 мар 2023, 17:59:59

до финиша: 22д.

...

Положение участников Задачи Посылки

19. Хипуй

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Выход	стандартный вывод или output.txt

В этой задаче вам необходимо самостоятельно (не используя соответствующие классы и функции стандартной библиотеки) организовать структуру данных Heap для хранения целых чисел, над которой определены следующие операции: а) Insert(k) – добавить в Heap число k ; б) Extract достать из Heap наибольшее число (удалив его при этом).

Формат ввода

В первой строке содержится количество команд N ($1 \leq N \leq 100000$), далее следуют N команд, каждая в своей строке. Команда может иметь формат: "0 <число>" или "1", обозначающий, соответственно, операции Insert(<число>) и Extract. Гарантируется, что при выполнении команды Extract в структуре находится по крайней мере один элемент.

Формат вывода

Для каждой команды извлечения необходимо отдельной строкой вывести число, полученное при выполнении команды Extract.

Пример 1

Ввод	Выход
2	
0 10000	10000
1	

Пример 2

Ввод	Выход
14	345
0 1	4346
0 345	2435
1	365
0 4346	235
1	5
0 2435	1
1	
0 235	
0 5	
0 365	
1	
1	
1	
1	

- 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

Язык Python 3.11.2

Набрать здесь Отправить файл

1

[Отправить](#)

 осталось 100 попыток

[Предыдущая](#)

[Следующая](#)

 Помощь нет

Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

🕒 19 фев 2023, 17:01:05

старт: 13 фев 2023, 15:00:00

финиш: 13 мар 2023, 17:59:59

до финиша: 22д.

...

Объявления жюри

Положение участников Задачи Посылки

20. Пирамидальная сортировка

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Выход	стандартный вывод или output.txt

Отсортируйте данный массив. Используйте пирамидальную сортировку.

Формат ввода

Первая строка входных данных содержит количество элементов в массиве N , $N \leq 10^5$. Далее задаются N целых чисел, не превосходящих по абсолютной величине 10^9 .

Формат вывода

Выведите эти числа в порядке неубывания.

Пример 1

Ввод

1
1

Вывод

1

Пример 2

Ввод

2
3 1

Вывод

1 3

- 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

Язык Набрать здесь

осталось 100 попыток

Посылок нет