Условия задач

Количество элементов во всех структурах данных не превышает 10000, если это не указано особо.

Сайт:	Дистанционная подготовка
Курс:	Структуры данных
Условия задач:	Условия задач
Printed by:	Гость
Date:	Понедельник 19 Март 2018, 20:11

Список задач

Простой стек

Задача A. Простой стек

Реализуйте структуру данных "стек". Напишите программу, содержащую описание стека и моделирующую работу стека, 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 n
: Добавить в стек число n (значение n задается после команды). Программа должна вывести ok.

pop
: Удалить из стека последний элемент. Программа должна вывести его значение.

back
: Программа должна вывести значение последнего элемента, не удаляя его из стека.

size
: Программа должна вывести количество элементов в стеке.

clear
: Программа должна очистить стек и вывести ok.

exit
: Программа должна вывести bye и завершить работу.

Входные данные

Команды управления стеком вводятся в описанном ранее формате по 1 на строке.

Гарантируется, что набор входных команд удовлетворяет следующим требованиям: максимальное количество элементов в стеке в любой момент не превосходит 100, все команды pop и back корректны, то есть при их исполнении в стеке содержится хотя бы один элемент.

Выходные данные

Требуется вывести протокол работы со стеком, по 1 сообщению в строке

Примеры

входные данные
push 3 push 14 size clear push 1 back push 2 back pop

size pop size exit
выходные данные
ok ok 2 ok ok 1 ok 2 2 1 1 0 bye

Стек с защитой от ошибок

Задача B. Стек с защитой от ошибок

Реализуйте структуру данных "стек". Напишите программу, содержащую описание стека и моделирующую работу стека, 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 n
: Добавить в стек число n (значение n задается после команды). Программа должна вывести ok.

pop
: Удалить из стека последний элемент. Программа должна вывести его значение.

back
: Программа должна вывести значение последнего элемента, не удаляя его из стека.

size
: Программа должна вывести количество элементов в стеке.

clear
: Программа должна очистить стек и вывести ok.

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.

Входные данные
Вводятся команды управления стеком, по одной на строке

Выходные данные
Программа должна вывести протокол работы стека, по одному сообщению на строке

Примеры

входные данные
size push 1 size push 2 size push 3 size exit

выходные данные
0 ok 1 ok 2 ok 3 bye

Стек неограниченного размера

Задача C. Стек неограниченного размера

Реализуйте структуру данных "стек". Напишите программу, содержащую описание стека и моделирующую работу стека, 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 n
Добавить в стек число n (значение n задается после команды). Программа должна вывести ok.

pop
Удалить из стека последний элемент. Программа должна вывести его значение.

back
Программа должна вывести значение последнего элемента, не удаляя его из стека.

size
Программа должна вывести количество элементов в стеке.

clear
Программа должна очистить стек и вывести ok.

exit
Программа должна вывести bye и завершить работу.

Размер стека должен быть ограничен только размером доступной оперативной памяти. 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.

Входные данные
Вводятся команды управления стеком, по одной на строке

Выходные данные
Требуется вывести протокол работы стека, по одному сообщению на строке

Примеры

входные данные
push 3 push 14 size clear push 1 back push 2 back

pop
size
pop
size
exit

выходные данные

ok
ok
2
ok
ok
1
ok
2
2
1
1
0
bye

Простая очередь

Задача D. Простая очередь

Реализуйте структуру данных "очередь". 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 строчку. 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.

Гарантируется, что набор входных команд удовлетворяет следующим требованиям: максимальное количество элементов в очереди в любой момент не превосходит 100, все команды pop и front корректны, то есть при их исполнении в очереди содержится хотя бы один элемент.

Входные данные

Вводятся команды управления очередью, по одной на строке

Выходные данные

Требуется вывести протокол работы с очередью, по одному сообщению на строке

Примеры

входные данные
size push 1 size push 2 size push 3

size exit
выходные данные
0 ok 1 ok 2 ok 3 bye

Очередь с защитой от ошибок

Задача E. Очередь с защитой от ошибок

Реализуйте структуру данных "очередь". 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 строчку. 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, содержится 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.

Входные данные
Вводятся команды управления очередью, по одной на строке

Выходные данные
Требуется вывести протокол работы очереди, по одному сообщению на строке

Примеры

входные данные
push 1 front exit
выходные данные
ok 1 bye

входные данные
size push 1 size push 2 size push 3 size exit
выходные данные
0 ok 1 ok 2 ok 3 bye

Очередь неограниченного размера

Задача F. Очередь неограниченного размера

Реализуйте структуру данных "очередь". 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 строчку. 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.

Размер очереди должен быть ограничен только размером доступной оперативной памяти. Перед исполнением операций front и pop программа должна проверять, содержится 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.

Входные данные
Вводятся команды управления очередью, по одной на строке

Выходные данные
Требуется вывести протокол работы очереди, по одному сообщению на строке

Примеры

входные данные
push 1 front exit
выходные данные
ok 1

bye

входные данные
size push 1 size push 2 size push 3 size exit
выходные данные
0 ok 1 ok 2 ok 3 bye

Простой дек

Задача G. Простой дек

Реализуйте структуру данных "дек". 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
Добавить (положить) в начало дека новый элемент. Программа должна вывести ok.

push_back
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. Все операции pop_front, pop_back, front, back всегда корректны.

Входные данные
Вводятся команды управления деком, по одной на строке.

Выходные данные

Требуется вывести протокол работы дека, по одному сообщению на строке.

Примеры

входные данные
push_back 1 back exit
выходные данные
ok 1 bye

входные данные
size push_back 1 size push_back 2 size push_front 3 size exit
выходные данные
0 ok 1 ok 2 ok 3 bye

Дек с защитой от ошибок

Задача H. Дек с защитой от ошибок

Реализуйте структуру данных "дек". 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
Добавить (положить) в начало дека новый элемент. Программа должна вывести ok.

push_back
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.

Входные данные

Вводятся команды управления деком, по одной на строке.

Выходные данные

Требуется вывести протокол работы дека, по одному сообщению на строке

Примеры

входные данные
push_back 1 back exit
выходные данные
ok 1 bye

входные данные
size push_back 1 size push_back 2 size push_front 3 size exit
выходные данные
0 ok 1 ok 2 ok 3 bye

Дек неограниченного размера

Задача I. Дек неограниченного размера

Реализуйте структуру данных "дек". 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
Добавить (положить) в начало дека новый элемент. Программа должна вывести ok.

push_back
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.

Размер дека должен быть ограничен только размером доступной оперативной памяти. 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.

Входные данные

Вводятся команды управления деком, по одной на строке.

Выходные данные

Требуется вывести протокол работы дека, по одному сообщению на строке.

Примеры

входные данные
push_back 1 back exit
выходные данные
ok 1 bye

входные данные
size push_back 1 size push_back 2 size push_front 3 size exit
выходные данные
0 ok 1 ok 2 ok 3 bye