***Субботина Олеся 422-3 отсортированный массив***

1.

***int queue\_size(******char \*path);***

***Описание***

Функция ***queue\_size*** принимает в качестве аргумента указатель на символьный массив (т.е. строку). Затем она открывает файл с указанным путем, считывает целочисленное значение из файла с помощью функции fscanf() и сохраняет его в переменной n. Наконец, функция закрывает файл. В общем, эта функция считывает одно целое значение из файла, которое представляет собой размер очереди, и возвращает это значение в виде целого числа.

***Параметры:***

принимает в качестве аргумента char-указатель path, который представляет собой путь к файлу.

***Дополнительные аргументы***

Не принимает

***Возвращаемое значение***

В случае успеха, возвращает целочисленное значение n.

В случае неудачи, "Cannot open file".

2.

***int\* create\_queue(char \*path);***

***Описание***

Функция она выделяет динамическую память для целочисленного массива размером n с помощью функции calloc(). Затем с помощью цикла считываются n целых чисел из файла и сохраняются в выделенном массиве. Затем функция выполняет сортировку вставкой в массив, чтобы отсортировать его элементы в порядке возрастания (т.е. от наименьшего к наибольшему). Функция закрывает файл.  
В общем, эта функция считывает список целых чисел из файла, сортирует их в порядке возрастания с помощью алгоритма сортировки вставкой и возвращает указатель на отсортированный массив.

***Параметры:***

***Дополнительные аргументы***

Нет

***Возвращаемое значение***

В случае успеха, возвращает указатель на отсортированный массив.

В случае неудачи, "Cannot open file".

3.

***int\* insert(int\* arr, int size, int value);***

***Описание***

Функция создает новый целочисленный массив new\_array размером size + 1 с помощью calloc(), который устанавливает все элементы массива в 0. Затем она просматривает исходный массив arr и находит индекс, в который должно быть вставлено значение, чтобы сохранить отсортированный порядок очереди. Он копирует элементы arr до точки вставки в new\_array. Затем вставляет значение в точку вставки и копирует оставшиеся элементы из arr в new\_array. Наконец, он освобождает память, выделенную для исходного массива, используя free, устанавливает его указатель на адрес new\_array и возвращает его.

В общем, эта функция вставляет новый элемент в отсортированный целочисленный массив и возвращает указатель на измененный массив.

***Параметры:***

три аргумента: указатель на целочисленный массив arr, целочисленное значение size, представляющее текущий размер массива, и целочисленное значение value, представляющее элемент, который нужно добавить в массив.

***Дополнительные аргументы***

нет

***Возвращаемое значение***

массив.

4.

***int\* delete\_max(int\* arr, int size)***

***Описание***

Эта функция delete\_max удаляет из массива элемент с максимальным значением и возвращает измененный массив. Вот как она работает в деталях:  
  
Сначала с помощью функции calloc() создается новый массив с тем же размером, что и исходный массив, минус один. Этот новый массив будет содержать измененное содержимое исходного массива после удаления максимального значения. Далее элементы исходного массива копируются в новый массив с помощью цикла for. Однако цикл останавливается на size-1, а не на size, так как в новом массиве на один элемент меньше. После того как новый массив заполнен содержимым исходного массива, функция free() используется для чистки памяти, выделенной под исходный массив.

***Параметры:***

указатель на целочисленный массив arr, целочисленное значение size, представляющее текущий размер массива

***Дополнительные аргументы***

Нет

***Возвращаемое значение***

массив.

5.

***int\* delete(int\* arr, int size, int value)***

***Описание***

Функция delete удаляет из массива первое вхождение элемента с указанным значением и возвращает измененный массив. Вот как она работает в деталях:

Сначала функция вызывает функцию поиска, чтобы найти индекс первого вхождения элемента с заданным значением. Если элемент не найден, функция возвращает исходный массив без каких-либо изменений. Если элемент найден, создается новый массив размером 1 с помощью функции calloc(). Этот новый массив будет содержать измененное содержимое исходного массива после удаления элемента. Затем используется цикл for для копирования элементов исходного массива до индекса первого вхождения указанного значения в новый массив. Затем используется еще один цикл for для копирования элементов после индекса первого вхождения указанного значения в новый массив, но индекс нового массива сдвигается на одну позицию назад. После того как новый массив заполнен содержимым исходного массива, функция free() используется для чистки памяти, выделенной под исходный массив.

***Параметры:***

указатель на целочисленный массив arr, целочисленное значение size, представляющее текущий размер массива и значение для поиска

***Дополнительные аргументы***

Нет

***Возвращаемое значение***

массив.

6.

***int search(int\* arr, int size, int value)***

***Описание***

Эта функция сначала инициализирует левую и правую границы диапазона поиска первым и последним индексами массива соответственно. Затем она многократно вычисляет средний индекс диапазона поиска и сравнивает значение по этому индексу с целевым значением. Если значение по среднему индексу равно целевому значению, функция возвращает средний индекс. Если значение по среднему индексу меньше целевого значения, функция обновляет левую границу на один индекс вправо от среднего индекса. Если значение среднего индекса больше целевого значения, функция обновляет правую границу на один индекс влево от среднего индекса. Функция продолжает этот процесс до тех пор, пока не будет найдено целевое значение или не будет исчерпан диапазон поиска.

В целом, эта функция обеспечивает эффективный способ поиска элемента с заданным значением в отсортированном целочисленном массиве с помощью двоичного поиска.

***Параметры:***

указатель на целочисленный массив arr, целочисленное значение size, представляющее текущий размер массива

***Дополнительные аргументы***

Нет

***Возвращаемое значение***

В случае успеха, возвращает Целое число(индекс).

В случае неудачи, целое число (-1).

7.

***int\* change\_value(int\* arr, int size, int*** ***changeable, int*** ***to\_change)***

***Описание***

Функция сначала удаляет элемент со значением to\_change из массива с помощью функции delete, которая не предусмотрена в данном коде. Затем она уменьшает размер массива и вставляет значение changeable в массив с помощью функции insert, которая также не предусмотрена в данном коде. Наконец, функция возвращает измененный массив.

В целом, эта функция пытается изменить значение выбранного элемента целочисленного массива, удаляя элемент со старым значением и вставляя элемент с новым значением.

***Параметры:***

указатель на целочисленный массив arr, целочисленное значение size, представляющее текущий размер массива и значения, на которое меняют changeable и которое меняют to\_change.

***Дополнительные аргументы***

Нет

***Возвращаемое значение***

массив.