Лабораторная работа №2

Тема. Структура данных массив. Указатели и массивы. Функции.

Цель. Приобретение навыков по выполнению операций над массивом в языке С++.

Получение навыков по разработке функций

**Задание 1.** Разработайте приложение для выполнения операций со статическим массивом.

1. Разработать функции для заполнения и отображения значений массива.
2. Разработать функции для выполнения всех задач варианта. Выполнить декомпозицию задач варианта и оформите подзадачи в виде отдельных функций.
3. Разработать программу. Программа должна содержать меню, позволяющее продемонстрировать работу всех операций над массивом.
4. Отладить программу. Привести в конце модуля программы тесты.

**Задание 2.** Разработайте программу для выполнения всех операций варианта над динамическим массивом.

**Задание 3.** Подготовьте отчет по выполненной работе по статическому и динамическому массивам. Ответьте на вопросы данной лабораторной работы. Включите в отчет описание алгоритма сортировки на естественном языке. Пример оформления отчета приведен в конце данного документа.

|  |  |  |  |
| --- | --- | --- | --- |
| вариант | Тип элемента массива | Операции над элементами структуры | Метод сортировки |
| 1 | Целый, с диапазоном значений -255..255 | 1. Вставить новый элемент в массив в позицию первого вхождения числа, сумма цифр которого кратна 7. 2. Найти количество элементов массива, являющихся простыми числами. 3. Удалить каждый элемент массива, который расположен после числа, цифры которого образуют упорядоченную по возрастанию последовательность. | Метод прямого обмена |
| 2 | Целый, с диапазоном значений -215..215-1 | 1. Вставить новое значение в массив перед каждым числом, цифры которого образуют арифметическую последовательность. 2. Выполнить циклический сдвиг всех элементов массива влево на М позиций. 3. Удалить каждый элемент, который расположен перед числом, в записи которого больше двух цифр равных заданной. | Метод прямой вставки |
| 3 | Целый, с диапазоном 0..255 | 1. Вставить новое значение в позицию каждого элемента, которое содержит число, являющееся степенью двух. 2. Найти последнее вхождение в массив числа, у которого равны первая и последняя цифры. 3. Расположить все числа массива, которые больше первого числа в массив, перед первым, а все меньшие и равные ему, после него. | Метод прямого выбора |
| 4 | Целый , с диапазоном 0..216 | 1. Вставить новое значение перед каждым элементом, которое содержит число, делящееся на каждую из своих цифр. 2. Определить, сколько раз входит в массив максимальное значение массива. 3. Удалить все числа массива, которые являются совершенными числами (число равно сумме своих делителей кроме самого числа: 6, 28). | Шейкер сортировка |
| 5 | Целый, с диапазоном 0..232-1 | 1. Вставить новый элемент перед каждым элементом, у которого четное количество цифр. 2. Удалить все числа массива, следующие за простым числом. 3. Найти максимальное число среди четных чисел массива. | Центрированная сортировка |
| 6 | Целый, с диапазоном значений -255..255 | 1. Вставить новый элемент после каждого элемента, в котором нет цифр равных нулю. 2. Удалить число, которое расположено после числа, которое делится на каждую из своих цифр. 3. Сформировать новый массив из индексов тех чисел массива, которые являются дружественными числами (сумма делителей одного равна сумме делителей другого). | Сортировка Шелла |
| 7 | Целый, с диапазоном значений -215..215-1 | 1. Вставить новый элемент перед каждым элементом, цифровой корень которого равен 7. 2. Удалить число массива перед числом, у которого сумма цифр кратна 7 и само число делится на 7. 3. Сколько чисел массива являются взаимно - простыми, т.е. наибольший общий делитель этих чисел равен 1. | Цифровая сортировка. |
| 8 | Целый, с диапазоном 0..255 | 1. Вставить значение 0 перед числом, наибольшая цифра которого – это первая цифра числа. 2. Удалить каждое число, которое расположено после числа, имеющего больше трех простых множителей. 3. Найти и вывести все числа, которые являются числами Армстронга (сумма цифр, возведе**нных** в степень n, равна самому числу) | Сортировка методом деления пополам |
| 9 | Целый , с диапазоном 0..216 | 1. Вставить число -1 после каждой упорядоченной последовательности из К чисел массива. 2. Удалить каждое число, расположенное непосредственно перед числом палиндромом. 3. Найти количество чисел, у которых цифра старшего разряда числа встречается в числе более одного раза. | Сортировка методом слияния |
| 10 | Целый, с диапазоном значений от 0 до 232-1 | 1. Вставить число -1 перед каждым числом, в котором только нечетные цифры. 2. Удалить число, расположенные непосредственно перед числами, в которых нет цифры 3. 3. Определить, сколько раз максимальное число массива встречается в массиве. | Сортировать по правилу:  с младших индексов расположить положительные, затем отрицательные затем нулевые. |
| 11 | Целый, с диапазоном значений -215..215-1 | 1. Вставить в позиции, указанные пользователем (в массиве индексов массива), значения, равные индексу, умноженному на 10. 2. Удалить числа массива в четных позициях, если они расположены между числами с четным и нечетным числом цифр. 3. Определить, образуют ли числа массива последовательность чисел Фибоначчи, в которой первое и второе число равно единице. | Сортировка по правилу: все элементы большие первого элемента массива разместить перед ним, затем все меньшие и равные первому. При перестановке сохранить порядок следования элементов в обеих частях массива. |
| 12 | Целый, с диапазоном 0..255 | 1. Вставить значение 0 перед элементами, находящимися в позиции, соответствующие числам Фибоначчи. 2. Удалить числа из позиций, которые являются простыми числами. 3. Найти самую длинную последовательность в массиве, которая состоит только из четных чисел. | В начало массива переместить те числа, которые начинаются с 1, затем те, которые начинаются с 2 и т.д. |
| 13 | Целый , с диапазоном 0..216 | 1. Вставить максимальное значение массива после элемента, у которого первая и последняя цифры равны. 2. Удалить элементы массива, цифры которых образуют последовательность чисел Фибоначчи, в которой первое и второе число равно 1. 3. Сформировать массив из индексов элементов массива, которые равны минимальному значению массива. | Сортировка распределением |
| 14 | Целый, с диапазоном значений -255..255 | 1. Вставить минимальное значение массива перед каждым элементом массива, у которого нечетных цифр больше чем четных. 2. Удалить все элементы массива, индексы которых ввел пользователь. 3. Сформировать массив индексов локальных максимумов (элемент, который больше предыдущего и больше следующего). | Сортировка методом двоичных деревьев |
| 15 | Целый, с диапазоном значений -215..215-1 | 1. Вставить перед каждым элементом, у которого первая цифра больше последней, число равное полусумме максимального и минимального значений массива. 2. Удалить каждый элемент, у которого все цифры равны. 3. Определить, упорядочен ли массив по возрастанию или по убыванию. | Сортировка подсчетом |
| 16 | Целый, с диапазоном 0..255 | 1. Вставить число равное среднему арифметическому чисел массива, которые разместились в массиве между максимальным и минимальным значениями, после каждого четного числа. 2. Удалить повторяющиеся элементы массива, оставив только первое вхождение каждого значения. 3. Сформировать массив из тех чисел исходного массива, двоичный код которых содержит три единицы. | Сортировать по правилу:  с младших индексов расположить четные, затем нечетные, затем нулевые. |
| 17 | Целый , с диапазоном 0..216 | 1. Вставить число, равное минимальному числу среди четных чисел массива, перед каждым нечетным числом массива. 2. Удалить все числа массива, в троичном коде которых ровно две единицы. 3. Определить, сколько раз в массив входит минимальное число массива (проход по массиву один раз). | Метод сортировки пузырьком с флагом |

**Пример разработки приложения и оформления отчета по заданию 1**

1. **Условие задачи**

Дан массив из n элементов целого типа long.

Сформировать массив из чисел Армстронга (153=13+53+33)

1. **Математическая модель**

Сумма возведенных в степень цифр числа. Степень равна количеству цифр в числе.

1. **Декомпозиция.**

Список подзадач

1. Определение количества цифр в числе
2. Возведение целого числа в степень
3. Определение числа Армстронга
4. **Определение функций**

***Ввод значений***

Предусловие. n – число заполняемых элементов,0 ≤n≤Max, где MAX – максимальное число элементов, L – массив с переменной верхней границей.

Постусловие. Заполненный массив из n элементов

void input\_ar(Array\_Static L[], int n);

Вывод значений массива

Предусловие. n>0

Постусловие. Вывод значений массива

void output\_ar(Array\_Static L[], int n);

*//Операции для решения поставленной задачи*

***Определение количества цифр в числе***

Предуслвие. n≥10

Постусловие. Результат целое, сумма цифр

int count(long n);

***Возведение целого числа в степень***

Предуслвие. а>0 – цифра числа, n>0 – количество цифр в числе

Постусловие. Результат целое, возведение а в степень n

long double pow\_1(unsigned short a, unsigned short n);

***Определение числа Армстронга***

Предуслвие. n>0 целое число,

Постусловие. Результат целое, сумма цифр

long double Armstrong(long n);

1. **Реализация функций**

**Примечание.** Следует прокомментировать используемые имена

//ввод в массив значений

void input\_ar(long L[],int n)

{ cout<<”введите”<<n<<”целых чисел”

for(int i=0;i<n;i++)

cin>>L[i];

}

//Вывод массива

void output\_ar(long L[], int n)

{

cout<<"Массив "<< endl;

for (int i=0;i< n; i++) {cout<<"a["<<i<<"]="<<L [i]<<endl;

}

и т. д. все операторы .

1. **Разработка алгоритма программы и представление его на псевдокоде.**
2. **Кодирование алгоритма программы**

#include"iostream"

const int Max= 100;

void input\_ar(long L[],int n);

void output\_ar(long L[], int n);

int main(int argc, char\* argv[])

{ long a[Max];

cout<<”Введите количество обрабатываемых элементов”;

cin>>n;

if (n<0 || n>Max) return 1;

input\_ar(a,n);

output\_ar(a,n);

cin.get();

return 0;

}

//ввод в массив значений

void input\_ar(long L[],int n)

{ cout<<”введите”<<n<<”целых чисел”

for(int i=0;i<n;i++)

cin>>L[i];

}

//Вывод массива

void output\_ar(long L[], int n)

{ cout<<"Массив "<< endl;

for (int i=0;i< n; i++)

{cout<<"a["<<i<<"]="<<L [i]<<endl;}

1. **Таблица тестов для каждой функции и программы.**

**Пример разработки приложения и оформления отчета по заданию 2**

1. **Определение функций**

Создание динамического массива из n элементов целого типа.

long \*create\_array(int &n);

Предусловие Нет

Постусловие Создает массив из n элементов целого типа. Возвращает указатель на массив из n и n, либо NULL, если память под массив не выделена.

Добавление нового элемента в массив

void insert\_elems(long \*&L,int &n,long x);

Предусловие. Массив должен существовать, т.е. L отлично от NULL.

Постусловие. Увеличивает размер массива на один элемент, добавляя новую ячейку в конец массива. Используется функция realloc из malloc.h

Удаление последней ячейки из массива

void delete\_elems(long \*&L,int &n);

Предусловие. Массив должен существовать.

Пострусловие. Из массива удаляется последняя ячейка. Размер массива уменьшается.

Добавление нового элемента в массив

void insert(long \*&L,int &n, long x);

Предусловие. Массив должен существовать, т.е. L отлично от NULL.

Постусловие. Увеличивает размер массива на один элемент, добавляя новую ячейку в конец массива. Используется функция memcpy из string.h

Ввод значений

void input\_ar(long L[], int n);

Предусловие. n – число заполняемых элементов,0 ≤n≤Max, где MAX – максимальное число элементов, L – массив с переменной верхней границей.

Постусловие. Заполненный массив из n элементов

Вывод значений массива

void output\_ar(long L[], int n);

Предусловие. n>0

Постусловие. Вывод значений массива

*//Операции для решения поставленной задачи*

Определение количества цифр в числе

int count(long n);

Предуслвие. n≥10

Постусловие. Результат целое, сумма цифр

Возведение целого числа в степень

long double pow\_1(unsigned short a, unsigned short n);

Предуслвие. а>0 – цифра числа, n>0 – количество цифр в числе

Постусловие. Результат целое, возведение а в степень n

Определение числа Армстронга

long double Armstrong(long n);

Предуслвие. n>0 целое число,

Постусловие. Результат целое, сумма цифр

1. **Реализация функций приложения**

#include "stdafx.h"

#include"iostream.h"

#include"malloc.h"

#include"string.h"

long \*create\_array(int &n);

void output\_ar(long L[], int n);

void input\_ar(long L[],int n);

void insert\_elems(long \*&L,int &n,long x);

void delete\_elems(long \*&L,int &n);

void insert(long \*&L,int &n,long x);

int main(int argc, char\* argv[])

{int n;

long \*a=create\_array(n);

input\_ar(a, n);

int x=10;

//insert\_elems( a, n,x);

//output\_ar(a, n);

delete\_elems(a,n); output\_ar(a, n);

insert(a,n,x);output\_ar(a, n);

cout<<(sizeof(a)/sizeof(long))<<endl;

return 0;

}

long \*create\_array(int &n)

{

cout<<"Óêàæèòå ðàçìåð ìàññèâà n=";

cin>>n;

long \*ptr=new long[n];

return ptr;

}

void input\_ar(long L[],int n)

{ cout<<"введите "<<n<<"элементов ";

for(int i=0;i<n;i++)

cin>>L[i];

}

void output\_ar(long L[], int n)

{ cout<<"Ìàññèâ "<< endl;

for (int i=0;i< n; i++)

cout<<"a["<<i<<"]="<<L [i]<<endl;

}

void insert(long \*&L,int &n,long x) //вставка нового значения в массив

{

memcpy(L,L,n+1); //увеличение размера массива –функция из string.h

//увеличение памяти по указателю L

L[n]=x;n++;

}

void insert\_elems(long \*&L,int &n,long x) //вставка нового значения в массив

{

realloc(L,(n+1)\*sizeof(long)); //увеличение размера массива –функция из malloc.h

//увеличение памяти по указателю L

L[n]=x;n++;

}

void delete\_elems(long \*&L,int &n) //удаление элемента из массива

{

realloc(L,(n-1)\*sizeof(long)); // изменение размера памяти по указателю L

n--;

**}**

Вопросы

* 1. Охарактеризуйте основные этапы жизненного цикла приложения.
  2. В чем состоит суть декомпозиции задачи?
  3. Каково назначение метода пошаговой детализации?
  4. Что определяет тип данных?
  5. Дайте определение структуре данных.
  6. Что значит *статическая структура данных*?
  7. Что значит *динамическая структура данных*?
  8. Раскройте понятия: *абстрактный тип данных, реализация АТД*.

# Пример разработки приложения и оформления отчета по заданию 2

1. **Определение функций**

Создание динамического массива из n элементов целого типа.

long \*create\_array(int &n);

Предусловие Нет

Постусловие Создает массив из n элементов целого типа. Возвращает указатель на массив из n и n, либо NULL, если память под массив не выделена.

Добавление нового элемента в массив

void insert\_elems(long \*&L,int &n,long x);

Предусловие. Массив должен существовать, т.е. L отлично от NULL.

Постусловие. Увеличивает размер массива на один элемент, добавляя новую ячейку в конец массива. Используется функция realloc из malloc.h

Удаление последней ячейки из массива

void delete\_elems(long \*&L,int &n);

Предусловие. Массив должен существовать.

Пострусловие. Из массива удаляется последняя ячейка. Размер массива уменьшается.

Добавление нового элемента в массив

void insert(long \*&L,int &n, long x);

Предусловие. Массив должен существовать, т.е. L отлично от NULL.

Постусловие. Увеличивает размер массива на один элемент, добавляя новую ячейку в конец массива. Используется функция memcpy из string.h

Ввод значений

void input\_ar(long L[], int n);

Предусловие. n – число заполняемых элементов,0 ≤n≤Max, где MAX – максимальное число элементов, L – массив с переменной верхней границей.

Постусловие. Заполненный массив из n элементов

Вывод значений массива

void output\_ar(long L[], int n);

Предусловие. n>0

Постусловие. Вывод значений массива

*//Операции для решения поставленной задачи*

Определение количества цифр в числе

int count(long n);

Предуслвие. n≥10

Постусловие. Результат целое, сумма цифр

Возведение целого числа в степень

long double pow\_1(unsigned short a, unsigned short n);

Предуслвие. а>0 – цифра числа, n>0 – количество цифр в числе

Постусловие. Результат целое, возведение а в степень n

Определение числа Армстронга

long double Armstrong(long n);

Предуслвие. n>0 целое число,

Постусловие. Результат целое, сумма цифр

1. **Реализация функций приложения**

#include "stdafx.h"

#include"iostream.h"

#include"malloc.h"

#include"string.h"

long \*create\_array(int &n);

void output\_ar(long L[], int n);

void input\_ar(long L[],int n);

void insert\_elems(long \*&L,int &n,long x);

void delete\_elems(long \*&L,int &n);

void insert(long \*&L,int &n,long x);

int main(int argc, char\* argv[])

{int n;

long \*a=create\_array(n);

input\_ar(a, n);

int x=10;

//insert\_elems( a, n,x);

//output\_ar(a, n);

delete\_elems(a,n); output\_ar(a, n);

insert(a,n,x);output\_ar(a, n);

cout<<(sizeof(a)/sizeof(long))<<endl;

return 0;

}

long \*create\_array(int &n)

{

cout<<"Óêàæèòå ðàçìåð ìàññèâà n=";

cin>>n;

long \*ptr=new long[n];

return ptr;

}

void input\_ar(long L[],int n)

{ cout<<"введите "<<n<<"элементов ";

for(int i=0;i<n;i++)

cin>>L[i];

}

void output\_ar(long L[], int n)

{ cout<<"Ìàññèâ "<< endl;

for (int i=0;i< n; i++)

cout<<"a["<<i<<"]="<<L [i]<<endl;

}

void insert(long \*&L,int &n,long x) //вставка нового значения в массив

{

memcpy(L,L,n+1); //увеличение размера массива –функция из string.h

//увеличение памяти по указателю L

L[n]=x;n++;

}

void insert\_elems(long \*&L,int &n,long x) //вставка нового значения в массив

{

realloc(L,(n+1)\*sizeof(long)); //увеличение размера массива –функция из malloc.h

//увеличение памяти по указателю L

L[n]=x;n++;

}

void delete\_elems(long \*&L,int &n) //удаление элемента из массива

{

realloc(L,(n-1)\*sizeof(long)); // изменение размера памяти по указателю L

n--;

**}**

**Вопросы**

* 1. Охарактеризуйте основные этапы жизненного цикла приложения.
  2. В чем состоит суть декомпозиции задачи?
  3. Каково назначение метода пошаговой детализации?
  4. Что определяет тип данных?
  5. Дайте определение структуре данных.
  6. Что значит *статическая структура данных*?
  7. Что значит *динамическая структура данных*?
  8. Раскройте понятия: *абстрактный тип данных, реализация АТД*.