Лабораторная Работа №0. Список Заданий

- Цель: научиться использовать функции, тренировка работы с массивами
- Для всех заданий необходимо написать функцию и тестирующую программу, которая **вводит исходные данные, вызывает функцию и выводит результат её работы**.
- Функция принимает на вход значения и возвращает результат их обработки, при этом функция не выводит на экран результат своей работы. Результат работы функции должен быть выведен на экран в функции main() (тестирующая программа).
- Название функции должно отражать ее функционал и начинаться с глагола, например, get, set, delete, erase, sum, sub и т.д.
- Разрешены к использованию следующие функции: printf, scanf, getchar (считывание символа с клавиатуры).

В Примере 4 программа вводит 10 вещественных чисел и подсчитывает их сумма с помощью функции.

Пример 4: Функция суммирования элементов массива типа float.

```c
#include <stdio.h>

#define N 10

float sum_array(float* array, int len) // функция
{
    int i = 0;
    float sum = 0.0;
    for(i = 0; i < len; i++)
    {
        sum += array[i];
    }
    return sum;
}

int main() // тестирующая программа
{
    float array[N];
    int i = 0;
    float s = 0.0;
    for(i = 0; i < N; i++)
    {
        scanf("%f", &array[i]);
    }
    s = sum_array(array, N);
    printf("sum =  %f\n", s);
```

```
    return 0;
}
```

Задания:

1. Написать функцию, которая из двух массивов типа int, упорядоченных по убыванию, формирует новый массив двойной длины, упорядоченный по убыванию (слияние).
2. Написать функцию нахождения в массиве целых чисел элемента, ближайшего к значению второго аргумента типа int.
3. Написать функцию нахождения в массиве целых чисел наименьшего по абсолютной величине числа.
4. Написать функцию нахождения в массиве вещественных чисел числа с наименьшей дробной частью (дробная часть всегда положительна).
5. Написать функцию нахождения в массиве целых чисел разности индексов максимального и минимального элементов.
6. Написать функцию, которая в массиве вещественных чисел обнуляет все элементы, которые меньше среднего арифметического значения элементов исходного массива.
7. Написать функцию, которая вставляет в массив элемент с заданным индексом и заданным значением. Лишний элемент должен пропасть.
8. Написать функцию, которая удаляет из массива элемент с заданным индексом. Недостающий элемент должен быть обнулен.
9. Написать функцию, которая переставляет элементы массива типа int так, что все положительные элементы предшествуют отрицательным.
10. Написать функцию, которая переставляет элементы массива типа int так, что все четные значения предшествуют нечетным.
11. Написать функцию, находящую максимум из значений четырех аргументов типа float.
12. Написать функцию, которая удаляет из массива все элементы, являющиеся локальными минимумами. Локальным минимумом считается элемент, который меньше и своего левого соседа, и своего правого соседа. Недостающие элементы должны быть обнулены.
13. Написать функцию, которая возводит первый аргумент в степень, равную второму аргументу. Все значения имеют тип int.
14. Написать функцию, удаляющую лидирующие и заключительные пробелы и символы табуляции.
15. Написать функцию, которая переставляет элементы массива типа int так, что первое значения меняется с последним, второе - с предпоследним, и т.д. Общее количество обменов определяется вторым аргументом.
16. Написать функцию, проверяющую, является ли первый аргумент некоторой натуральной степенью второго аргумента. Все значения имеют тип int.
17. Написать функцию, находящую наименьшее общее кратное (НОК) двух чисел.
18. Написать функцию, которая удаляет из массива все элементы, являющиеся локальными максимумами. Локальным максимумом считается

элемент, который больше и своего левого соседа, и своего правого соседа. Недостающие элементы должны быть обнулены.

19. Написать функцию, которая в массиве вещественных чисел обнуляет все элементы, которые больше среднего арифметического значения элементов исходного массива.
20. Написать функцию нахождения в массиве целых чисел наибольшего по абсолютной величине числа.

Лабораторная Работа №1. Список Заданий

- Цель: работа с массивами строк, работа с указателями.
- При работе с массивами необходимо использовать синтаксис указателей (char *s; s++; whie(*s != 0)).
- Необходимо использовать двумерный символьный массив для представления строки в качестве набора слов (каждое слово хранится в отдельной строке двумерного массива).
- В данной лабораторной работе можно использовать допущение о максимальной длине строки, чтобы не использовать динамическую память.
- Программа должна состоять из нескольких функций.
- Разрешены к использованию следующие функции: printf, scanf, getchar (считывание символа с клавиатуры), gets, fgets.

В Примере 5 функция разбирает строку на слова (отделены пробелами) и записывает результат в двумерный массив.

Пример 5: Функция заполнения двумерного массива слов на языке C.

```c
char parts[100][256];
int str_convert(char *s, char parts[][256])
{
    int r = 0;
    while(*s)
    {
        if(*s != ' ')
        {
            char *ptr = parts[r++];
            do
            {
                *(ptr++) = *(s++);
            } while(*s && *s != ' ');
            *ptr = '\0';
        }
        else
            s++;
```

```
    }
    return r;
}
```

Задания:

1. Ввести строку. Вывести слова в алфавитном порядке.
2. Ввести строку. Вывести пословно в порядке убывания длин слов.
3. Ввести строку. Вывести пословно в порядке возрастания количества гласных букв.
4. Ввести строку. Вывести только слова, в которых нет повторяющихся букв.
5. Ввести строку. Вывести слова, в которых каждая буква входит не менее двух раз.
6. Ввести строку. Вывести её, удалив предварительно повторяющиеся слова.
7. Ввести строку. Вывести различные слова вместе с количеством их появления в строке
8. Ввести строку. Вывести только различные встречающиеся целые числа.
9. Ввести строку и слово. Определить и вывести, сколько раз встречаются в строке буквы, перечисленные в слове.
10. Ввести строку и слово. Вывести строку, удалив из него все вхождения слова.
11. Ввести строку и слово. Вывести порядковые номера слов в строке, совпадающих с введенным словом
12. Ввести строку. Вывести строку так, чтобы за каждым словом следовало количество пробелов, равное длине слова.
13. Ввести строку и слово. Вывести количество слов в строке, имеющих длину, равную длине введенного слова.
14. Ввести строку. Вывести её, заменив цифры на слова (0 - "zero", 1 - "one", ...).
15. Ввести строку и слово. Вывести те слова, которые не содержат букв, входящих в слово.
16. Ввести строку, вывести на экран пословно, причем каждое слово вывести в обратном порядке.
17. Ввести строку и слово, удалить из строки все слова, которые лексикографически меньше, чем введенное слово.
18. Ввести строку и слово, вывести все слова, которые содержат введенное слово как составную часть.
19. Ввести строку. Вывести только различные встречающиеся отрицательные действительные числа.
20. Ввести строку. Вывести строку так, чтобы за каждым словом следовало количество пробелов, равное номеру слова в строке.

Лабораторная Работа №2. Список Заданий

- Во всех заданиях с использованием массива (в том числе на полиномы) массив необходимо выделить динамически с помощью malloc

1. Ввести степень и коэффициенты полинома. Ввести границы интервала a, b и точность eps. Найти корень полинома на интервале [a, b] методом деления отрезка пополам (считать, что p(a) * p(b) < 0).
2. Вычислить значение определенного интеграла от функции cos(x) на интервале [0.5, 0.8] по методу трапеций с точностью eps.
3. Ввести матрицу размером 5x5 и найти все её седловые точки (вывести их индексы). Седловая точка - это максимум по строке и минимум в столбце или наоборот.
4. Ввести массив вещественных чисел. Вывести максимальную длину серии локальных экстремумов (минимумов или максимумов).
5. Ввести длину массива натуральных чисел и диапазон его значений (минимально возможное и максимально возможное значения). Ввести массив. Вывести упорядоченный массив при помощи сортировки подсчётом: для каждого возможного значения массива заводится свой счётчик; каждый новый элемент массива увеличивает соответствующий его значению счётчик на 1; на экран выводятся все значения из возможного диапазона в количестве, соответствующем значению их счётчиков.
6. Сортировка вставками: пусть первые k элементов упорядочены по возрастанию. Берется (k+1)-ый элемент и размещается среди первых k так, чтобы упорядоченными оказались k+1 элементов. Этот метод применяется при k от 1 до n-1.
7. Рекурсивно описать функцию C(n, m) вычисления биномиального коэффициента по следующей формуле: C(n, 0) = C(n, n) = 1; C(n, m) = C(n - 1, m) + C(n - 1, m - 1) при 0 < m < n. Написать тестирующую программу к ней.
8. Ввести длину массива и массив. Упорядочить массив по возрастанию методом быстрой сортировки: Выбрать средний элемент массива и переставить элементы так, чтобы слева оказались только меньшие, а справа - только большие, чем средний. После этого рекурсивно применить этот метод к левой и правой частям.
9. Ввести массив целых чисел и отсортировать его (можно использовать qsort). Ввести число и найти в массиве ближайшее к нему методом двоичного поиска.
10. Написать функцию двоичного поиска в упорядоченном массиве слов.
11. Ввести два упорядоченных массива. Получить упорядоченный массив путем слияния двух введенных и вывести его.
12. Ввести массив вещественных чисел. Вывести все локальные максимумы в порядке возрастания значений и глобальный минимум.
13. Ввести массив вещественных чисел. Вывести локальный экстремум (минимум или максимум), ближайший к нулю.

14. Ввести массив целых чисел. Вывести позиции максимального и минимального значений скользящей суммы из 5-ти соседних элементов. Для крайних элементов использовать циклическое замыкание.
15. Ввести массив вещественных чисел. Найти и вывести номер элемента, для которого сумма разностей с соседними элементами максимальна. Для крайних элементов использовать циклическое замыкание.
16. Ввести длину массива и массив вещественных чисел. Найти и вывести отрезок из 5 элементов с максимальным средним арифметическим значением. Для крайних элементов использовать циклическое замыкание.
17. Ввести массив целых чисел. Отсортировать его по возрастанию двоичных весов элементов.
18. Ввести матрицу размером 5x5 и найти все точки, являющиеся одновременно максимумом и в своей строке, и в своем столбце (вывести их индексы).
19. Ввести степень и коэффициенты многочлена. Ввести число k. Вывести k-ю производную введенного многочлена.
20. Ввести массив вещественных чисел. Вывести минимальную длину серии чисел, не являющихся локальными экстремумами (минимумами или максимумами).

Лабораторная Работа №3. Список Заданий

Шаблоны и исключения
Во всех задачах требуется реализовать конструктор копий и оператор присваивания. Если в задании не оговорено ограничение на максимальный размер, то считается, что контейнер динамически растет.
Во всех заданиях стоит избегать избыточного копирования параметров.

В main должно быть продемонстрировано, что шаблонный класс работает:

- для любого примитивного типа
- **для struct Point { int x; int y; int z; }**

- 1. Сделать реализацию vector на шаблонах.
  - vector(size_t size, const T& value)
  - push_back
  - size
  - operator[]
  - at (бросает исключения)
  - insert(size_t i, const T& value) (бросает исключения)
  - erase(size_t i) (бросает исключения)
- 2. Сделать реализацию list (классы node и list) на шаблонах.
  - push_back
  - push_front
  - size
  - find_and_erase(const T& value)
  - at (бросает исключения)
- 3. Сделать реализацию двусвязного списка double_list (классы node и double_llist) на шаблонах.
  - push_back
  - push_front
  - size
  - find_and_erase(const T& value)
  - at (бросает исключения)
- 4. Сделать реализацию матрицы на шаблонах. Методы set, get, sum бросают исключения.
  - matrix(size_t row, size_t column)
  - set
  - get
  - rows() --- количество строк
  - columns() --- количество столбцов
  - matrix sum(const matrix& m)

- 5. Сделать реализацию матрицы на шаблонах. Методы set, get, mul бросают исключения.
  - matrix(size_t row, size_t column, const T& value)
  - set
  - get
  - rows() --- количество строк
  - columns() --- количество столбцов
  - matrix mul(const matrix& m)
- 6. Сделать реализацию отсортирванного массива sorted_array с фиксированной capacity (не растет) на шаблонах. При добавлении элемента массив остается отсортированным. Для реализации с Point использовать перегрузку оператора меньше.
  - sorted_array(size_t capacity)
  - push (бросает исключения при превышении capacity)
  - size
  - operator[]
  - at (бросает исключения)
  - erase(size_t i) (бросает исключения)
- 7. Сделать реализацию очереди queue на шаблонах.
  - enque
  - T deque() (бросает исключение, если очередь пуста)
  - size
  - clear
  - operator<< для вывода
- 8. Сделать реализации стека stack на шаблонах.
  - push
  - T pop() (бросает исключение, если очередь пуста)
  - size
  - clear
  - operator<< для вывода
- 9. Сделать реализацию множества set (на основе массива с сортировкой) для хранения только уникальных элементов.
  - insert(const T& value)
  - erase(const T& value) (бросает исключение на пустом множестве)
  - bool find(const T& value)
  - size
  - clear
  - operator<< для вывода
- 10. Сделать реализацию множества set (на основе массива с сортировкой) для хранения только уникальных элементов.
  - insert(const T& value)
  - bool find(const T& value)
  - size() возвращает количество элементов в множестве

- set union(const set& s) --- объединение (бросает исключение при объединении с пустым множеством)
- operator<< для вывода

- 11. Сделать реализацию множества set (на основе массива с сортировкой) для хранения только уникальных элементов.
  - insert(const T& value)
  - bool find(const T& value)
  - size() возвращает количество элементов в множестве
  - set intersect(const set& s) --- пересечение (бросает исключение при пересечении с пустым множеством)
  - operator<< для вывода

Лабораторная Работа №4. Список Заданий

- Используйте приватные поля для данных.
- Реализуйте конструктор, который должен устанавливать значения по умолчанию,
- Реализуйте методы для чтения и записи значений полей (set, get)
- Если требуется, напишите конструктор копий, оператор присваивания и деструктор.
- Реализуйте перегруженные операторы, указанные задания
- Ввод, вывод полей класса должен быть реализован через перегрузку операторов operator<< и operator>>
- Напишите программу для тестирования вашего класса и всех операций

1. Создайте класс BinPolynom (полином с двоичными коэффициентами) с операциями +, +=, -, -=, =, ==, !=, >, <, >=, <=, *, *=. Все операции над коэффициентами выполняются в двоичной арифметике.
2. Создайте класс Date с операциями +(int), +=(int), =, ==, !=, >, <, >=, <=, ++ и --. (int) - количество дней.
3. Создайте класс Time с операциями +(int), +=(int), =, ==, !=, >, <, >=, <=, ++, --. (int) - секунды.
4. Создайте класс Polynom (многочлен) с операциями +, +=, -, -=, =, ==, !=, >, <, >=, <=, *, *=.
5. Создайте класс BinVect (двоичный вектор) с операциями +, +=, -, -=, =, ==, !=, >, <, >=, <=, << (int) (сдвиг влево), >> (int) (сдвиг вправо).

6. Создайте класс Rational (рациональная дробь) с операциями +, +=, -, -=, =, ==, !=, >, <, >=, <=, *, *=, /, /=.
7. Создайте класс IntMod (операции по модулю простого числа) с операциями +, +=, -, -=, =, ==, !=, >, <, >=, <=, *, *=, /, /=.
8. Создайте класс String с операциями +, +=, =, ==, !=, >, <, >=, <=.
9. Создайте класс IntSet (множество целых) с операциями +, +=, -, -=, =, ==, !=, >, <, >=, <=, *, + (int) (добавить в множество), - (int) (исключения из множества).
10. Создайте класс IntHuge (длинные целые, представленные массивом разрядов) с операциями +, +=, -, -=, =, ==, !=, >, <, >=, <=, *, *=, *(int) (умножение на константу).
11. Создайте классы StrList (линейный упорядоченный список слов) с операциями += (char*) (включить в список), -= (char*) (исключить из списка), + (StrList) (слияние списков), ==, !=, >, >=, <, <=, ==(char*), != (char*).

Лабораторная Работа №5.

Необходимо реализовать многопоточное клиент-серверное приложение под Linux.

- Клиент - программа, запускаемая из консоли.
- Сервер - демон, корректно завершающийся по сигналам SIGTERM и SIGHUP.
- Клиент должен передать содержимое текстового файла через TCP.
- Сервер должен принять и сохранить в файл.