**Лабораторная работа №6**

**Рекурсивные функции**

17 вариант

1. Дано натуральное число N. Вычислите произведение его цифр.

Для решения задачи составить: не рекурсивную функцию, рекурсивную функцию, возвращающую значение, рекурсивную функцию типа **void**. Объединить функции в одной программе и сравнить результаты.

1. В одномерном массиве вещественных чисел, вычислить отдельно произведение положительных и отрицательных элементов, затем сравнить модули произведений и вывести сообщение о том, какое из них больше.

Функции ввода и вывода массива сделать рекурсивными без использования циклов.

**Текст программы**

1 задание (произведение цифр числа):

#include <iostream>

using namespace std;

int p=1;

int not\_recursion(int n) {

while (n > 0) {

p \*= n % 10;

n /= 10;

}

return p;

};

int int\_recursion(int n) {

p \*= n % 10;

n /= 10;

if (n > 0) return int\_recursion(n);

else return p;

};

void void\_recursion(int n) {

p \*= n % 10;

n /= 10;

if (n > 0) void\_recursion(n);

else cout << p << endl;

};

int main() {

setlocale(LC\_ALL, "Russian");

int n, f;

cout << "Введите число: "; cin >> n;

cout << "Введите номер функции: "; cin >> f;

switch (f) {

case 1: cout << not\_recursion(n) << endl; break;

case 2: cout << int\_recursion(n) << endl; break;

case 3: void\_recursion(n); break;

default: cout << "Такой функции нет!" << endl;

}

system("pause");

return 0;

}

2 задание (ввод и вывод массива):

#include <iostream>

#include <time.h>

using namespace std;

int i = 0;

void array\_input(int n, double\* a) {

a[i] = rand() / 100.0 - 50;

i++;

if (i < n) array\_input(n, a);

else i = 0;

};

void array\_output(int n, double\* a) {

cout << a[i] << " ";

i++;

if (i < n) array\_output(n, a);

else i = 0;

};

int main() {

setlocale(LC\_ALL, "Russian");

srand(time(0));

int n, kp = 0, ko = 0;

double p=1, o=1;

cout << "Введите количество элементов массива: "; cin >> n;

double\* a = new double[n];

array\_input(n, a);

array\_output(n, a);

cout << endl;

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

if (a[i] > 0) {

p \*= a[i];

kp++;

}

else if (a[i] < 0) {

o \*= a[i];

ko++;

}

}

if (kp == 0) p = 0;

if (ko == 0) o = 0;

cout << "Произведение положительных элементов: " << p << endl;

cout << "Произведение отрицательных элементов: " << o << endl;

o = fabs(o);

if (p > o) cout << "Модуль произведения положительных чисел больше" << endl;

else if (p < o) cout << "Модуль произведения отрицательных чисел больше" << endl;

else cout << "Модули произведений положительных и отрицательных чисел одинаковы" << endl;

delete[] a;

system("pause");

return 0;

}

**Описание программы**

1 задание (произведение цифр числа):

1. Задается переменная p, которой присваивается значение 1
2. Программа запрашивает число n, произведение цифр которого нужно найти
3. Программа запрашивает номер функции f, с помощью которой нужно найти произведение числа
4. Оператор switch: программа выполняет выбранную функцию

* Case 1: вывод функции not\_recursion (функция без рекурсии)
* Case 2: вывод функции int\_recursion (функция типа int с рекурсией)
* Case 3: выполнение функции void\_recursion (функция типа void с рекурсией)
* Default: вывод фразы «Такой функции нет!»

Описание функций

not\_recursion – не рекурсивная функция:

1. Цикл: пока n>0, переменная p умножается на остаток от деления n на 10, затем n делится нацело на 10
2. Результатом функции является получившееся произведение p

int\_recursion – рекурсивная функция типа int:

1. Переменная p умножается на остаток от деления n на 10
2. Переменная n делится нацело на 10
3. Если n>0, заново выполняется функция int\_recursion, иначе результатом функции является получившееся произведение p

void\_recursion – рекурсивная функция типа void:

1. Переменная p умножается на остаток от деления n на 10
2. Переменная n делится нацело на 10
3. Если n>0, заново выполняется функция void\_recursion, иначе выводится получившееся произведение p

2 задание (ввод и вывод массива):

1. Задается переменная i, которой присваивается значение 0.
2. Программа запрашивает у пользователя количество элементов динамического массива
3. Программа создает одномерный динамический массив a[n]
4. Программа вызывает функцию array\_input
5. Программа вызывает функцию array\_output
6. Программа проверяет, является ли каждый из элементов массива положительным или отрицательным
7. Программа отдельно вычисляет произведение положительных и отрицательных чисел
8. Программа выводит на экран получившиеся произведения положительных и отрицательных чисел
9. Программа находит модуль произведения отрицательных чисел
10. Программа сравнивает произведение положительных чисел с модулем произведения отрицательных чисел
11. Программа выводит сообщение о том, какое из 2 произведений больше (или сообщение о том, что они равны)

Описание функций

array\_input – ввод массива:

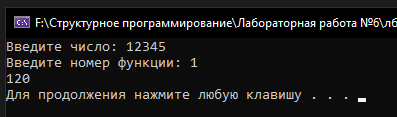
1. Элемент массива a[i] заполняется рандомным числом
2. Переменная i увеличивается на 1
3. Если i<n, заново выполняется функция array\_input, иначе переменная i приравнивается нулю

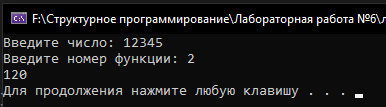
array\_output – вывод массива:

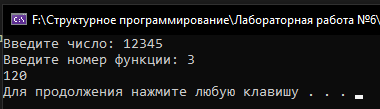
1. Выводится элемент массива a[i]
2. Переменная i увеличивается на 1
3. Если i<n, заново выполняется функция array\_input, иначе переменная i приравнивается нулю

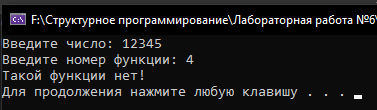
**Результаты тестов**

1 задание (произведение цифр числа):

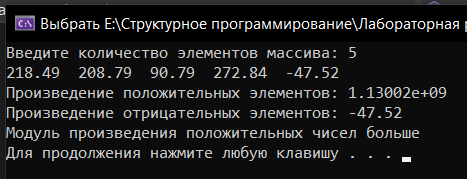




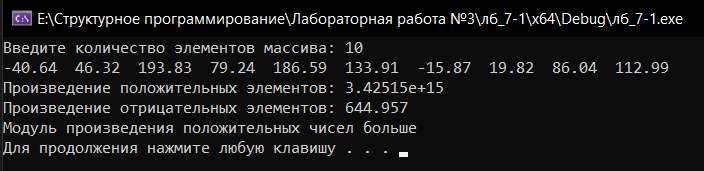




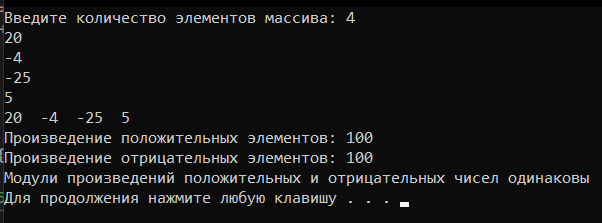
2 задание (ввод и вывод массива):



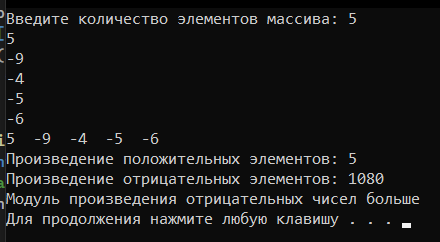




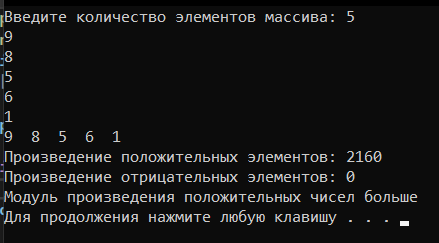














**Вывод**

Мною были написаны 2 программы с использованием рекурсивных функций.

Первая программа считает произведение цифр числа. Делает она это с помощью трех вариантов функций: не рекурсивную, рекурсивную типа int и рекурсивную типа void.

Вторая программа находит произведения положительных и отрицательных элементов одномерного динамического массива по отдельности, а затем сравнивает их модули. При этом ввод и вывод массива реализованы с использованием рекурсивных функций array\_input и array\_output.