###### Лабораторная работа №6

**Рекурсивные функции**

**Задание 1**

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

1. Найти сумму: 22+42+62+82+… (*n* слагаемых)

**Задание 2**

В задании№1 Лабораторной работы №1 (этот семестр) функции ввода и вывода массива сделать рекурсивными без использования циклов.

**Задача 1**

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

1. Найти сумму: 22+42+62+82+… (*n* слагаемых)

**Код программы:**

#include <iostream>

using namespace std;

int unrec(int n) {

int summ = 0;

for (int i = 1; i <= n; i++) {

summ += pow(2 \* i, 2);

}

return summ;

}

int rec\_i(int n) {

if (n == 0) {

return 0;

}

else {

return 4 \* pow(n, 2) + rec\_i(n - 1);

}

}

void rec\_v(int n, int &result) {

if (n == 0) ;

else {

result += 4 \* pow(n, 2);

rec\_v(n - 1, result);

}

}

int main() {

setlocale(LC\_ALL, "Russian");

int n;

cout << "Введите n: ";

cin >> n;

cout << "Нерекурсивная: " << unrec(n) << endl;

cout << "Рекурсивная (int): " << rec\_i(n) << endl;

int result = 0;

cout << "Рекурсивная (void): ";

rec\_v(n, result);

cout << result << endl;

if ((unrec(n) == rec\_i(n)) and (unrec(n) == result)) {

cout << "Результаты совпадают" << endl;

}

else {

cout << "Результаты не совпадают" << endl;

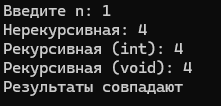
}

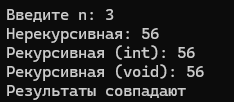
return 0;

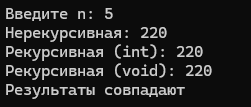
system("pause");

}

**Примеры работы программы:**

****

****

****

**Задача 2**

В задании№1 Лабораторной работы №1 (этот семестр) функции ввода и вывода массива сделать рекурсивными без использования циклов.

**Код программы:**

#include <iostream>

#include <string>

#include <windows.h>

using namespace std;

//Структура, содержащая информацию о поезде

struct TRAIN {

string name;

string date;

string type;

};

void input\_array\_rec(int i, TRAIN\* mas, int n) {

if (i == n) {

return;

}

cout << "-" << i + 1 << "-\n";

cout << "Название: ";

cin >> mas[i].name;

cout << "Тип: ";

cin >> mas[i].type;

cout << "Дата отправления: ";

cin >> mas[i].date;

cout << endl;

input\_array\_rec(i + 1, mas, n);

}

void find\_by\_type\_rec(int i, TRAIN\* mas, int n, string info, int& count) {

if (i == n) {

if (count == 0) {

cout << "Таких поездов нет." << endl;

}

return;

}

if (info == mas[i].type) {

cout << mas[i].name << endl;

cout << mas[i].type << endl;

cout << mas[i].date << endl;

cout << endl;

}

else {

count--;

}

find\_by\_type\_rec(i + 1, mas, n, info, count);

}

int main() {

//Установка кодировки консоли

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "Богданов Ренат. Структуры." << endl;

cout << "Описание поездов." << endl << endl;

cout << "Введите информацию о поездах:" << endl << endl;

//Количество структур TRAIN в массиве

const int n = 5;

//Счётчик подходящих поездов

int count = n;

//Тип поезда

string info;

//Инициализация массива структур TRAIN

TRAIN mas[n];

input\_array\_rec(0, mas, n);

//Ввод типа поезда

cout << "Введите тип поезда (пассажирский, товарный, почтовый): ";

cin >> info;

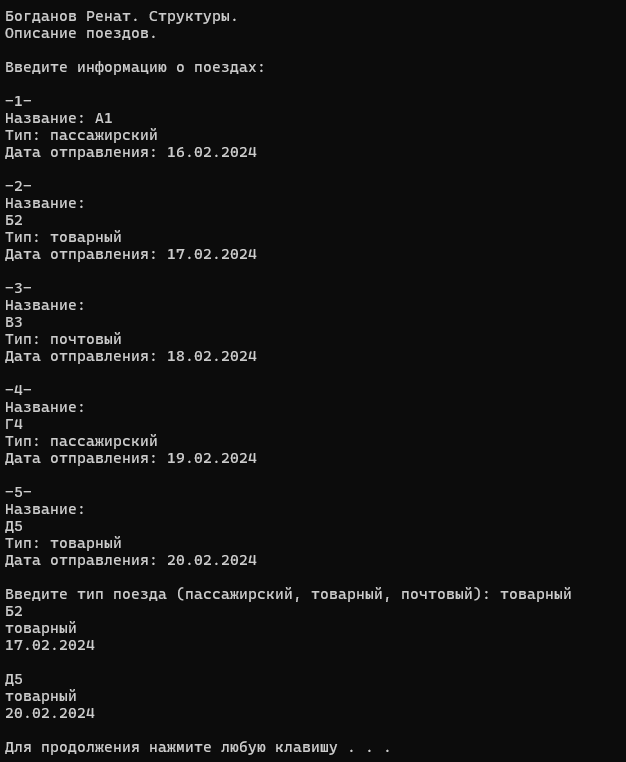
find\_by\_type\_rec(0, mas, n, info, count);

system("pause");

return 0;

}

**Результаты работы программы:**

****