Лабораторная работа №6

Рекурсивные функции.

ЗАДАНИЕ

Вариант 6

1. Составить:
   1. Не рекурсивную функцию.
   2. Рекурсивную функцию, возвращающую значение.
   3. Рекурсивную функцию типа void.
2. Объединить функции в одной программе и сравнить результаты.

Найти значение выражения: *an*.

ХОД РАБОТЫ

Результат работы программы представлен в соответствии с рисунком 1.

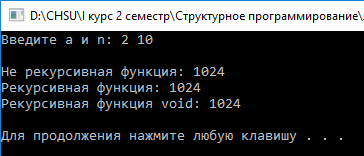


Рисунок 1 – результаты работы программы.

Текст программы:

#include <iostream>

using namespace std;

int rNot(int a, int n)

{

int s = a;

while (n > 1) {

s \*= a;

n--;

}

return s;

}

int rInt(int a, int n)

{

if (n == 1) return a;

else return rInt(a, 1) \* rInt(a, n-1);

}

void rVoid(int a, int n, int& s) {

if (n == 1) return;

s \*= a;

rVoid(a, n-1, s);

}

void main() {

setlocale(LC\_CTYPE, "rus");

int a, n, s;

cout << "Введите a и n: ";

cin >> a >> n;

cout << endl << "Не рекурсивная функция: " << rNot(a, n);

cout << endl << "Рекурсивная функция: " << rInt(a, n);

s = a;

rVoid(a, n, s);

cout << endl << "Рекурсивная функция void: " << s << endl;

cout << endl;

system("pause");

}

Алгоритм программы:

Пользователь вводит с клавиатуры значения a (целочисленная переменная a) и n (целочисленная переменная n).

Для не рекурсивной функции – объявление функции rNot с параметрами a и n, в результате которой возвращается значение a, возведенной в степень n.

Для рекурсивной функции с возвращением значения – объявление функции rInt. Внутри функции проверяется выражение n=1. Если условие верно, то возвращается значение a, если ложно, то функция вызывает себя с параметрами a и n-1. До тех пор, пока n не станет равным 1, результаты возвращений функций перемножаются между собой.

Для рекурсивной функции типа void – объявление функции rVoid по аналогии с функцией rInt, с ссылкой на переменную s в качестве дополнительного (третьего) параметра, чтобы сохранять в неё значение a в степени n. Для этого в функции main() объявим целочисленную переменную s. В функции проверяется выражение n=1. Если оно верно, то выполнение функции прекращается. В случае если оно ложно - умножается s на a и функция вызывает себя с параметрами: a, n-1, s.

Описать структуру AEROFLOT, содержащую поля: название пункта назначения, номер рейса, тип самолета. Реализовать ввод данных с клавиатуры в массив, состоящий из 5 структур типа AEROFLOT. Вывести на экран названия пунктов назначения и номера рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры. Если таких рейсов нет, то вывести соответствующее сообщение. В задании функции ввода и вывода массива сделать рекурсивными без использования циклов.

Результат работы программы представлен в соответствии с рисунком 1.

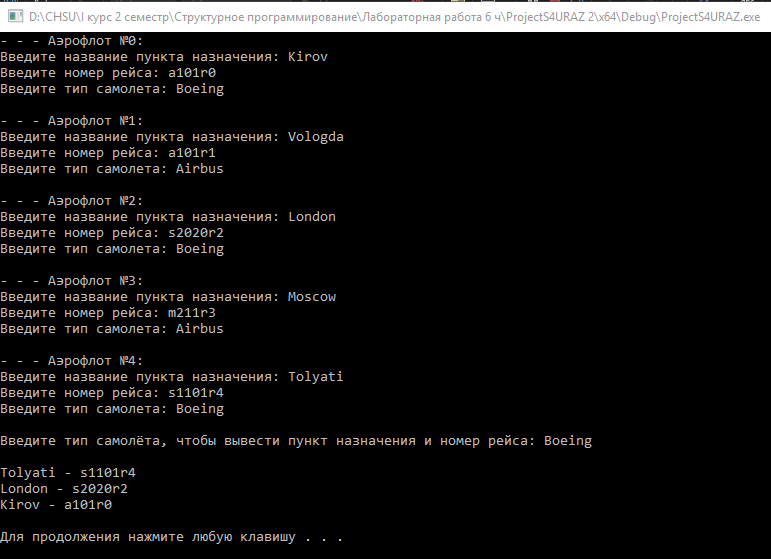


Рисунок 1 – результаты работы программы.

Текст программы:

#include <iostream>

#include <string>

using namespace std;

struct Aeroflot

{

char name[100];

char number[10];

char type[50];

};

int k = 0;

void enter(Aeroflot\* a, int n, int i)

{

if (i < n) {

cout << "- - - Аэрофлот №" << i << ":" << endl;

cout << "Введите название пункта назначения: ";

cin >> a[i].name;

cout << "Введите номер рейса: ";

cin >> a[i].number;

cout << "Введите тип самолета: ";

cin >> a[i].type;

cout << endl;

enter(a, n, i+1);

}

return;

}

void search(Aeroflot\* a, int n, char s[50])

{

if (n >= 0) {

if (!strcmp(s, a[n-1].type)) {

cout << a[n-1].name << " - ";

cout << a[n-1].number << endl;

k++;

}

search(a, n - 1, s);

}

else {

if (!k)

cout << "Тип самолёта не найден!" << endl;

return;

}

}

int main()

{

setlocale(LC\_CTYPE, "RUS");

const int n = 5;

Aeroflot a[n];

enter(a, n, 0);

char s[50];

cout << "Введите тип самолёта, чтобы вывести пункт назначения и номер рейса: ";

cin >> s;

cout << endl;

search(a, n, s);

cout << endl;

system("pause");

}

Алгоритм программы:

Описана структура Aeroflot: название пункта назначения (символьная переменная name), номер рейса (символьная переменная number) и тип самолета (символьная переменная type).

Объявлена глобальная переменная k, равная 0. Необходима в дальнейшем как счётчик количества найденных самолетов.

Объявление рекурсивной функции enter с тремя параметрами: указатель на первый объект структуры Aeroflot, значение количества элементов (переменной n), значение целочисленной переменной i. Внутри функции проверяется выражение i<n. Если условие верно, то пользователь с клавиатуры вводит для i объекта структуры значение переменных name, number, type. Затем функция вызывает себя с параметрами a, n, i+1. Если ложно, то функция завершает работу. Таким образом, с помощью рекурсивной функции enter, пользователем с клавиатуры задаются n объектов структуры Aeroflot.

Объявление рекурсивной функции search с тремя параметрами: указатель на первый объект структуры Aeroflot, значение количества элементов (переменной n), значение символьной переменной s (введенный для поиска тип самолета). Внутри функции проверяется выражение: n больше, либо равно 0.

Если выражение верно, то проверяется условие, что функция strcmp для сравнения символов двух строк возвращает значение 0 (строки равны). Если строка s и переменная type объекта n-1 структуры Aeroflot равны, то выводится переменные name и number этого объекта, а также к счётчику k прибавляется единица. Затем функция вызывает себя с параметрами a, n-1, s.

Если же выражение ложно, то проверяется условие, что счётчик равен 0. Если счётчик равен 0, то выводится информация о том, что тип самолёта не найден. Затем функция прекращает работу.

Объявлена целочисленная константа n, равная 5. Объявляется структура Aeroflot с n объектами. Вызывается функция enter с параметрами: a, n, 0. Пользователю предлагается ввести тип искомого самолета. Объявляется символьная переменная s, значением которой является введенный с клавиатуры тип искомого самолёта. Вызывается функция search с параметрами: a, n, s.

Вывод:

В ходе выполнения лабораторной работы мы научились использовать рекурсивные функции и закрепили полученные навыки.