МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЁТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ассистент | |  |  | | | |  | | Д. А. Кочин |
| должность, уч. степень, звание | |  | подпись, дата | | | |  | | инициалы, фамилия |
| ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №1 | | | | | | | | | | |
| Рекурсия | | | | | | | | | | |
| по дисциплине: ТЕОРИЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ | | | | | | | | | | |
|  | | | | | | | | | | |
| РАБОТУ ВЫПОЛНИЛА | | | | | | | | | | |
| СТУДЕНТКА ГР. | 4931 | | |  | 07.10.2022 |  | | Е.Ю. Ильченко | | |
|  |  | | |  | подпись, дата |  | | инициалы, фамилия | | |
|  |  | | |  |  |  | |  | | |

Санкт-Петербург 2022

1. **Цель работы:**

Цель настоящей работы – научиться преобразовывать арифметические функции в рекурсивные, используя операторы примитивной рекурсии, а также создавать функции для их вычислений на языке высокого уровня программирования.

1. **Основные сведения из теории:**

Рекурсия (от лат. recurso – бегу назад, возвращаюсь) – есть такой способ задания вычислимой функции, когда каждое ее значение задается через значение этой (или другой) вычислимой функции для меньших (или ранее определенных) значений аргументов, а функции задаваемые таким образом называются рекурсивными.

Все рекурсивные функции строятся на базе элементарных функций и операторов формирования из этих функций всё более и более сложных.

Элементарные функции:

* Нуль-функция (*On = 0*)
* Функция проекции (*Imn(x1, …, xm, …, xn) = xm*)
* Функция непосредственного следования (*S1(x) = x + 1*)

Операторы формирования:

* Оператор суперпозиции (*hn(x1, …, xn) = gm(f1n(x1, …, xn), …, fmn(x1, …, xn))*)
* Оператор примитивной рекурсии (*fn+1(x1, …, xn, y+1) = hn+2(x1, …, xn, y, fn(x1, …, xn))*)

1. **Постановка задачи:**

|  |  |
| --- | --- |
| **Вариант 39** | 3x - y |

Необходимо преобразовать заданную арифметическую функцию в рекурсивную, используя операторы примитивной рекурсии.

Во второй части лабораторной работы требуется создать программу на языке высокого уровня содержащую две функции. Первая функция должна вычислять заданную арифметическую функцию итеративным способом. Вторая функция должна вычислять  
заданную арифметическую функцию рекурсивным способом.

Аргументы арифметической функции должны задаваться пользователем в процессе  
выполнения программы;

1. **Преобразование арифметической функции в рекурсивную с использованием операторов примитивной рекурсии:**

Пусть заданы *On = 0* и *S1(x) = x + 1*.

Определим функции

*Sum(x, y) = S1(Sum(x, y-1)),* *Sum(x, 0) = x;*

*Mul3(x) = Sum(Mul3(x - 1), S1(S1(S1(O())))), Mul3(0) = 0;*

*Sub12(x, 0) = S1(Sub2(x-1, 0)), Sub12(0, 0) = 0;*

*Sub12(x, 1) = Sub12(x-1, 0), Sub12(1, 1) = 0;*

*Sub11(x) = Sub12(x, 1);*

*Sub(x, y) = Sub11(Sub(x, y-1)), Sub(x, 0) = x;*

Тогда искомая *F(x) = Sub(Mul3(x), y). F(x) = Sub(Mul3(x), y).*

1. **Листинг программы на языке высокого уровня с комментариями:**

#include <iostream>

using namespace std;

//итеаративная функция

int iterative(int x, int y) {

int counter1=x, counter2=y, result =0;

//Пока counter1 не 0 - добавляем каждый раз по 3 к результату

while (counter1 > 0) {

result += 3;

counter1--;

}

//Пока counter2 не 0 - уменьшаем результат на 1

while (counter2 > 0) {

result--;

counter2--;

}

return result;

}

//нуль функция

int nullfun() {

return 0;

}

//функция непосредственного следования

int nextfun(int x) {

return x + 1;

}

//Функция суммы двух чисел

int sum(int x, int y) {

if (y == nullfun()) {

return x;

}

else {

return nextfun(sum(x, y - 1));

}

}

//Функция 3\*x

int mul3(int x) {

if (x == nullfun()) {

return nullfun();

}

else {

return sum(mul3(x - 1), nextfun(nextfun(nextfun(nullfun()))));

}

}

//Функция вычитания единицы

int sub1(int x, int y=1) {

if (y == 0) {

if (x == 0) {

return nullfun();

}

else {

return(nextfun(sub1(x - 1, 0)));

}

}

else {

if (x == 1) {

return nullfun();

}

else {

return sub1(x - 1, 0);

}

}

}

//Функция вычитания двух чисел

int sub(int x, int y) {

if (y == 0) {

return x;

}

else {

return sub1(sub(x, y - 1));

}

}

//Искомая функция

int f(int x, int y) {

return sub(mul3(x), y);

}

int main()

{

setlocale(LC\_ALL, "ru");

system("color F1 ");

int x, y;

cout << "Функция 3x-y \nВведите x:\n";

cin >> x;

while (cin.fail() || cin.get() != '\n')

{

cin.clear();

cin.ignore(cin.rdbuf()->in\_avail(), '\n');

cout << "Неверный ввод, повторите еще раз:\n ";

cin >> x;

}

cout << "x = "<<x<<"\nВведите y:\n";

cin >> y;

while (cin.fail() || cin.get() != '\n')

{

cin.clear();

cin.ignore(cin.rdbuf()->in\_avail(), '\n');

cout << "Неверный ввод, повторите еще раз:\n ";

cin >> y;

}

cout << "y = " << y << "\n";

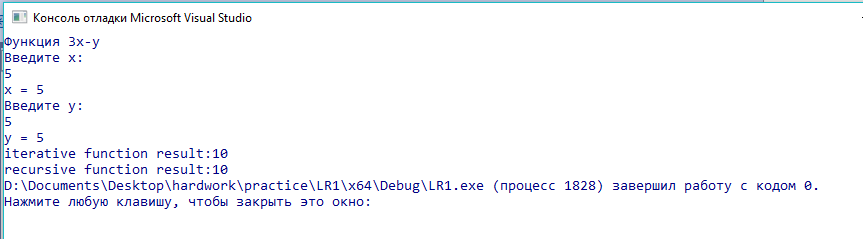
cout << "iterative function result:" << iterative(x,y);

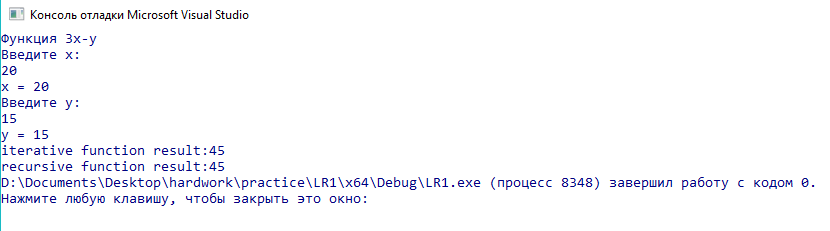
cout << "\nrecursive function result:" << f(x, y);

return 0;

}

1. **Пример результата выполнения:**

****

****

1. **Вывод:**

В ходе выполнения данной лабораторной работы были изучены элементарные функции и операторы формирования. Также была преобразована арифметическая функция в рекурсивную и реализована на языке высокого уровня. Кроме того, данная функция была реализована итеративно для проверки работоспособности рекурсивной функции