МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент Л.Н. Бариков

Отчёт

по лабораторной работе №6

по дисциплине ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: «Перегрузка функций»

Работу выполнил

студент гр. 4941 Н.С. Горбунов

Санкт-Петербург

2020

***Цель лабораторной работы:*** *изучение концепции подпрограмм и способов передачи аргументов в функции, освоение методов процедурного программирования на языке C++ при решении типовых вычислительных задач.*

***Задание на программирование:*** *используя технологию процедурного программирования, разработать программу, решающую поставленную задачу в соответствии с индивидуальным заданием.*

***Порядок выполнения работы:***

1. Получить у преподавателя индивидуальное задание и выполнить постановку задачи: сформулировать условие, определить входные и выходные данные, их ограничения.

2. Разработать математическую модель.

3. Построить схемы алгоритмов основных функций и задачи в целом. При этом необходимо учесть, что операция сравнения вещественных значений на равенство (если она необходима по условию задачи) не корректна. Например, пусть вычисляются два вещественных значения *alf*\_1 и *alf*\_2. Операция сравнения этих значений на равенство *alf*\_1==*alf*\_2 не корректна. Корректной операцией в этом случае будет операция |*alf*\_1-*alf*\_2|<*ɛ*, где *ɛ* - точность сравнения двух вещественных величин на равенство. Значение точности *ɛ* выбирается из диапазона (0, 0.1].

4. Составить программу на языке *C*++.

5. Входные данные на этапах тестирования и демонстрации работы преподавателю должны вводиться с клавиатуры по запросу. Выходные данные должны выводиться на экран с пояснениями.

6. Использовать стандартные потоковые объекты ввода/вывода ***cin*** и ***cout***.

7. Разработанная программа должна содержать функцию, выполняющую индивидуальное задание. При этом разработанная функция должна быть реализована 3 способами:

- как функция, возвращающая искомое значение;

- как функция, возвращающая искомое значение через дополнительный аргумент - указатель;

- как функция, возвращающая искомое значение через дополнительный аргумент - ссылку.

Эти три функции должны иметь одно и то же имя, т.е. должны быть перегружены.

Задача должна быть решена с использованием вызовов функций всех трех типов из функции *main*().

8. Оформить отчет о лабораторной работе в составе: постановка задачи, математическая модель, схемы алгоритмов, текст программы, контрольные примеры (скриншоты).

**Задание 13**

Разработать функцию, вычисляющую наименьшее общее кратное (НОК) двух натуральных чисел.

Ввести три натуральных числа. Используя разработанную функцию найти их НОК.

**Математическая модель**

НОК(a,b) = a\*b/НОД(a,b)

Для нахождения НОД двух натуральных чисел используется алгоритм Евклида. Классический алгоритм Евклида применяется к паре неотрицательных целых чисел. Пока ни одно из чисел в паре не равно нулю, из большего числа вычитается меньшее; вычитаемое и полученная разность формируют новую пару. Действие повторяется, пока один из элементов пары не обратится в 0, тогда значение другого будет равно искомому НОД.

Рекурсивная формулировка классического алгоритма:

НОД(a, 0) = a

НОД(a, b) = НОД(a, a − b) при a ≥ b

НОД(a, b) = НОД(b, b − a) при a < b

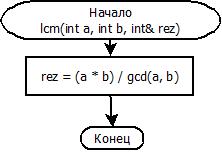
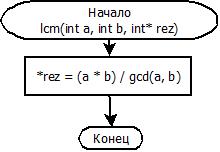
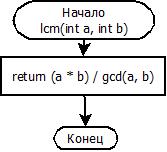
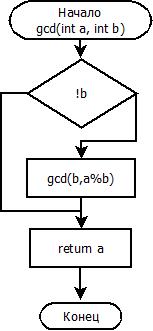
Несложно заметить, что последовательное вычитание из большего числа меньшего, пока разность не станет меньше вычитаемого, соответствует нахождению остатка от деления большего числа на меньшее. На этом основан так называемый быстрый алгоритм Евклида: в паре чисел одно число делится с остатком на второе; делитель и полученный остаток формируют новую пару. Действие повторяется, пока один из элементов пары не обратится в 0, тогда значение другого будет равно искомому НОД.

Рекурсивная формулировка быстрого алгоритма:

НОД(a, 0) = a

НОД(a, b) = НОД(b, ОСТАТОК(b / a))

**Блок-схема алгоритма**



**Текст программы**

#include <iostream>

using namespace std;

int gcd(int a, int b);

int lcm(int a, int b);

void lcm(int a, int b, int\* rez);

void lcm(int a, int b, int& rez);

int main()

{

int var, a, b, c, result;

setlocale(LC\_ALL,"Russian");

for (;;)

{

cout << "\n Вид действия:\n";

cout << " 1 - получение значений\n";

cout << " 2 - завершение работы программы\n";

cout << " Выберите вид действия -> ";

cin >> var;

switch (var)

{

case 1:

cout << "\nВведите три натуральных числа: ";

cin >> a >> b >> c;

cout << "\nФункция, возращающая значение \nНОК = " << lcm(a, lcm(b, c));

lcm(b, c, &result);

lcm(result, a, &result);

cout << "\nФункция с параметром-указателем\nНОК = " << result;

lcm(b, c, result);

lcm(a, result, result);

cout << "\nФункция с параметром-ссылкой\nНОК = " << result;

break;

default: return 0;

}

}

}

//функция возвращает НОД по бысртрому алгоритму Евклида

int gcd(int a, int b) {

return !b ? a : gcd(b, a % b); // если b не ноль, то вернуть а, иначе gcd(b, a % b)

}

// НОК двух натуральных чисел, функция возвращает значение.

int lcm(int a, int b)

{

return (a \* b) / gcd(a, b);

}

// функция передает значение через указатель

void lcm(int a, int b, int\* rez)

{

\*rez = (a \* b) / gcd(a, b);

}

// функция передает значение по ссылке

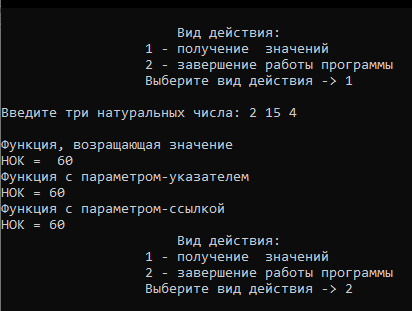
void lcm(int a, int b, int& rez)

{

rez = (a \* b) / gcd(a, b);

}

**Пример работы программы**

****

**Вывод:** используя технологию процедурного программирования, разработал программу, решающую поставленную задачу в соответствии с индивидуальным заданием.