Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**Дисциплина: Криптографические протоколы**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.Е.Гиренко

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Крамаренко

**Цель работы:** Реализовать программный продукт решения системы сравнений с помощью китайской теоремы об остатках с указанием промежуточных результатов (значения M, сравнения и решения сравнений первой степеней My=amodm). Программный продукт так же должен реализовывать возможность того, что система не имеет решений или имеет больше одного решения.

**Ход работы:**

1) Для начала напишем функцию ввода сравнений первого порядка при помощи двух коэффициентов:

void inputVectors() {

vector<pair<ll, ll>> input;

cout << "Введите либо 2 числа, коэффициенты а и б в выражении х=а mod б, либо ничего, чтобы закончить ввод строк" << endl;

int number = 1;

while (true) {

string s;

cout << number++ << ": ";

getline(cin, s);

vector<ll> koefs = separateStringToKoefs(s);

if (koefs.size() == 2) {

input.push\_back(make\_pair(koefs[0], koefs[1]));

}

else {

break;

}

}

solveChinaSystem(input);

}

2) В результате работы этой функции получаем вектор пар чисел, который будем использовать в основной функции хода решения системы уравнений. Так как по условиям задачи нас просят вывести промежуточные результаты, то после каждого шага выводим значения:

void solveChinaSystem(vector<pair<ll, ll>> input) {

int k = input.size();

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

cout << "x = " << input[i].first << " mod " << input[i].second << endl;

}

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

for (int j = i+1; j < k; j++) {

if (gcd(input[i].second, input[j].second) != 1) {

cout << "Нод всех коэффициентов б не равен 1. Решения нет" << endl;

return;

}

}

}

ll m = 1;

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

m \*= input[i].second;

}

cout << "M = " << m << endl;

vector<ll> ms(k);

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

ms[i] = m / input[i].second;

cout << "M"<<i+1 <<" = " << ms[i] << "; ";

}

cout << endl;

vector<ll> ns(k, 0);

vector<ll> f(k, 0);

ll c;

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

ns[i] = ms[i] % input[i].second;

ns[i] = pow(ns[i], euler(input[i].second) - 1);

ns[i] %= input[i].second;

cout << "N" << i + 1 << " = " << ns[i] << "; ";

}

cout << endl;

ll x = 0;

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

x += ns[i] \* ms[i] \* input[i].first;

}

x %= m;

cout << "Ответ: " << x << " + k\*" << m << endl;

}

3) В функции обязательно указываем проверку на наличие решения при помощи проверки, взаимно просты ли все модули:

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

for (int j = i+1; j < k; j++) {

if (gcd(input[i].second, input[j].second) != 1) {

cout << "Нод всех коэффициентов б не равен 1. Решения нет" << endl;

return;

}

}

}

4) Самую сложную часть, нахождение обратного элемента по модулю, находим при помощи функции Эйлера, так как вычислительные мощности компьютера позволяют нам считать такие большие числа:

ns[i] = pow(ns[i], euler(input[i].second) - 1);

В результате имеем программный продукт с удобным интерфейсом, способный решать систему уравнений с помощью китайской теоремы остатков при любом количестве решений.

**Листинг программы**

#include <iostream>

#include <chrono>

#include <random>

#include <vector>

#include <string>

#include <algorithm>

#include "windows.h"

#include <numeric>

using namespace std;

typedef long long ll;

ll euler(ll num) {

ll ret = 1, i, pow;

for (i = 2; i <= sqrt(num) && num != 1; i++)

{

pow = 1;

while (!(num % i))

{

num /= i;

pow \*= i;

}

ret \*= (pow - (pow / i));

}

if (num != 1)

ret \*= (num - 1);

return ret;

}

void solveChinaSystem(vector<pair<ll, ll>> input) {

int k = input.size();

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

cout << "x = " << input[i].first << " mod " << input[i].second << endl;

}

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

for (int j = i+1; j < k; j++) {

if (gcd(input[i].second, input[j].second) != 1) {

cout << "Нод всех коэффициентов б не равен 1. Решения нет" << endl;

return;

}

}

}

ll m = 1;

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

m \*= input[i].second;

}

cout << "M = " << m << endl;

vector<ll> ms(k);

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

ms[i] = m / input[i].second;

cout << "M"<<i+1 <<" = " << ms[i] << "; ";

}

cout << endl;

vector<ll> ns(k, 0);

vector<ll> f(k, 0);

ll c;

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

ns[i] = ms[i] % input[i].second;

ns[i] = pow(ns[i], euler(input[i].second) - 1);

ns[i] %= input[i].second;

cout << "N" << i + 1 << " = " << ns[i] << "; ";

}

cout << endl;

ll x = 0;

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

x += ns[i] \* ms[i] \* input[i].first;

}

x %= m;

cout << "Ответ: " << x << " + k\*" << m << endl;

}

vector<ll> separateStringToKoefs(string s) {

vector<ll> v;

int startInd = 0;

for (int i = 0; i < s.size() + 1; i++) {

if (s[i] == ' ' || i == s.size()) {

string subs = s.substr(startInd, i - startInd);

startInd = i + 1;

try {

v.push\_back(stol(subs));

}

catch (exception e) {}

}

}

return v;

}

void inputVectors() {

vector<pair<ll, ll>> input;

cout << "Введите либо 2 числа, коэффициенты а и б в выражении х=а mod б, либо ничего, чтобы закончить ввод строк" << endl;

int number = 1;

while (true) {

string s;

cout << number++ << ": ";

getline(cin, s);

vector<ll> koefs = separateStringToKoefs(s);

if (koefs.size() == 2) {

input.push\_back(make\_pair(koefs[0], koefs[1]));

}

else {

break;

}

}

solveChinaSystem(input);

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

inputVectors();

}