g***Министерство образования Республики Беларусь***

***Учреждение образования***

***«Брестский государственный технический университет»***

***Кафедра ИИТ***

**Лабораторная работа №4**

**По дисциплине КМЗИ за IV семестр**

**Тема: «Факторизация чисел»**

**Выполнил:**

Студент группы ИИ-15 (1)

2-го курса

Волк И. А.

**Проверил:**

Хацкевич М. В.

Брест 2019

Цель работы: изучить методы факторизации чисел.

Ход работы

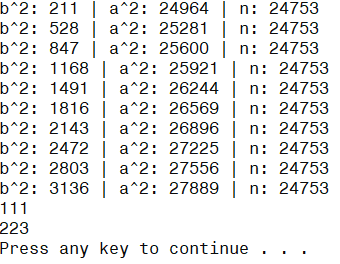
В данной лабораторной работе я реализую метод Ферма.

Реализованная программа состоит из 2 частей:

1. Основная программа реализующая метод (С++)
2. Скрипт для проверки, является ли число квадратном (Python)

Для простоты демонстрации будем использовать небольшие числа, хотя возможная работа с числами произвольной длины.

Вывод для числа 24753 выглядит следующим образом:



Листинги реализации:

1. Используемые заголовки:

#include <iostream>

#include <fstream>

#include <sstream>

#include <thread>

#include <boost/multiprecision/cpp\_int.hpp>

#include <boost/multiprecision/cpp\_dec\_float.hpp>

1. Фрагмент main:

int main()

{

auto dividers = FermatFactorization(mp::cpp\_int{

111\*223

}, 1000);

for (int i = 0; i < dividers.size(); i++)

std::cout << dividers[i] << std::endl;

system("pause");

}

Как видно, работа производится над числом 111\*223, а значит программа успешно справилась со своей задачей. Также стоит отметить, что работа выполняется посредством библиотеки boost/multiprecision, а значит работу производить можно с числами любой длины.

1. Реализация метода факторизации

std::vector< mp::cpp\_int> FermatFactorization(mp::cpp\_int n, int n0)

{

if (n <= 0)

return { n };

if (n % 2 == 0) return { 2, n / 2 };

mp::cpp\_int a = mp::sqrt(n) + 1;

for (int i = 0; i < n0; i++, a++)

{

mp::cpp\_int b = a\*a - n;

std::cout << "b^2: " << b

<< " | a^2: " << a \* a << " | n: " << n << std::endl;

if (isPerfectSquare(b))

return { a - mp::sqrt(b), a + mp::sqrt(b) };

}

return { n };

}

1. Функция, взаимодействующая со скриптом проверки числа на квадрат:

bool isPerfectSquare(mp::cpp\_int number)

{

// compose a command

// python perfect\_square.py <number>

std::stringstream ss;

ss << number;

std::string command = "python perfect\_square.py " + ss.str();

// call python script

system(command.c\_str());

std::this\_thread::sleep\_for(1s);

// get result

std::fstream fin("is\_perfect\_sqrt.tmp");

std::string result;

fin >> result;

fin.close();

std::remove("is\_perfect\_sqrt.tmp");

// estimate result

if (result == "true")

return true;

else if (result == "error")

{

std::cerr << "Error occurred while checking for perfect square" << std::endl;

return false;

}

else

return false;

}

1. Скрипт проверки, является ли число идеальным квадратом:

import math

import argparse

import sys

number = eval(sys.argv[1])

try:

root = int(math.sqrt(number))

except:

open('is\_perfect\_sqrt.tmp', 'w+').write('error')

if root \*\* 2 == number:

open('is\_perfect\_sqrt.tmp', 'w+').write('true')

else:

open('is\_perfect\_sqrt.tmp', 'w+').write('false')

Вывод: по ходу лабораторной работы научился производить проверку числа на простоту.