Звіт

**про виконання лабораторної роботи №2**

**Точна арифметика цілих чисел**

Виконав студент групи К-22

Коломієць Микола

1. **Довгі числа**

**1.0 Клас Number та основні відомості про нього**

У ході лабораторної роботи був створений клас довгого числа. За основне поле класу було обрано контейнер стандартної бібліотеки шаблонів deque. Його було обрано через швидке додавання нових елементів попереду контейнера. За додавання нулів попереду числа (задля вирівнювання двох чисел ( деякі алгоритми множення вимагають однакових за довжиною чисел)) відповідальність несе метод 0add, його часто буде видно в алгоритмах.

Поле mod відповідає за систему числення числа. В цій лабі реалізована робота лише з модулями меншими або рівними за 10. В алгоритмах використовуються числа лише з модулем 10 або 2 (у методах Кука або Шанхаге).

Поле с відповідає за знак числа (с = 1 додатне, с = 0 від’ємне).

Поле q – дробна частина числа (ця механіка не повністю реалізована в усіх методах проте у методах Кука працює).

Реалізовані основні оператори ++, +=, +, -, \*, %, /…

Присутня функція зчитування числа з командної строки (проте в алгоритмах вона не з’являлась)

Реалізовані деякі гетери (їх призначення інтуітивно можна зрозуміти з назви наприклад get\_q дістає дробову частину числа (до дробової частини є і сетер))

Реалізовані методи зміни модуля числа сhange\_mod2 та 10.

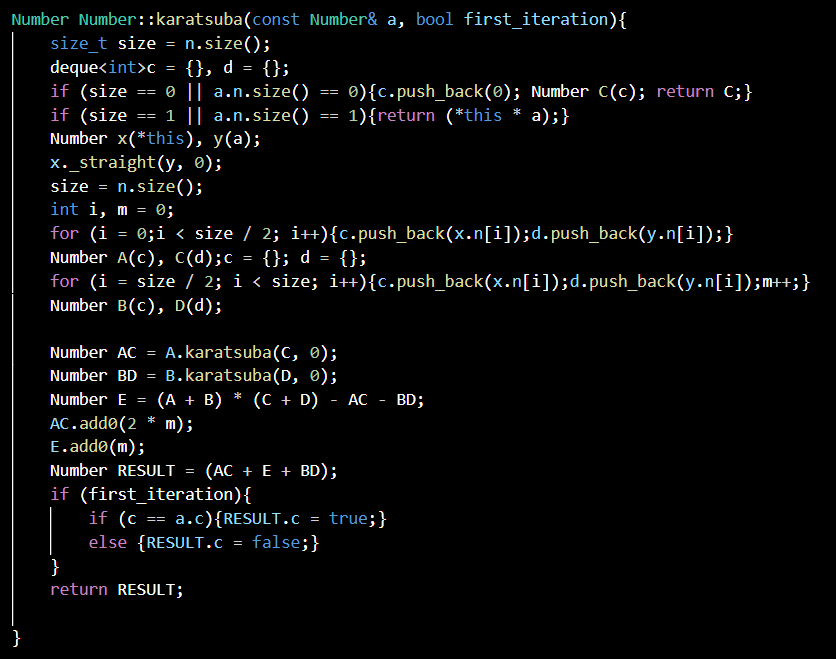
Реалізація виводу числа за допомогою cout.

Реалізований ввід чисел за допомогою допоміжної функції.

Задля більшої читабельності програми, вона була розділена на заголовочний файл та сpp файл.

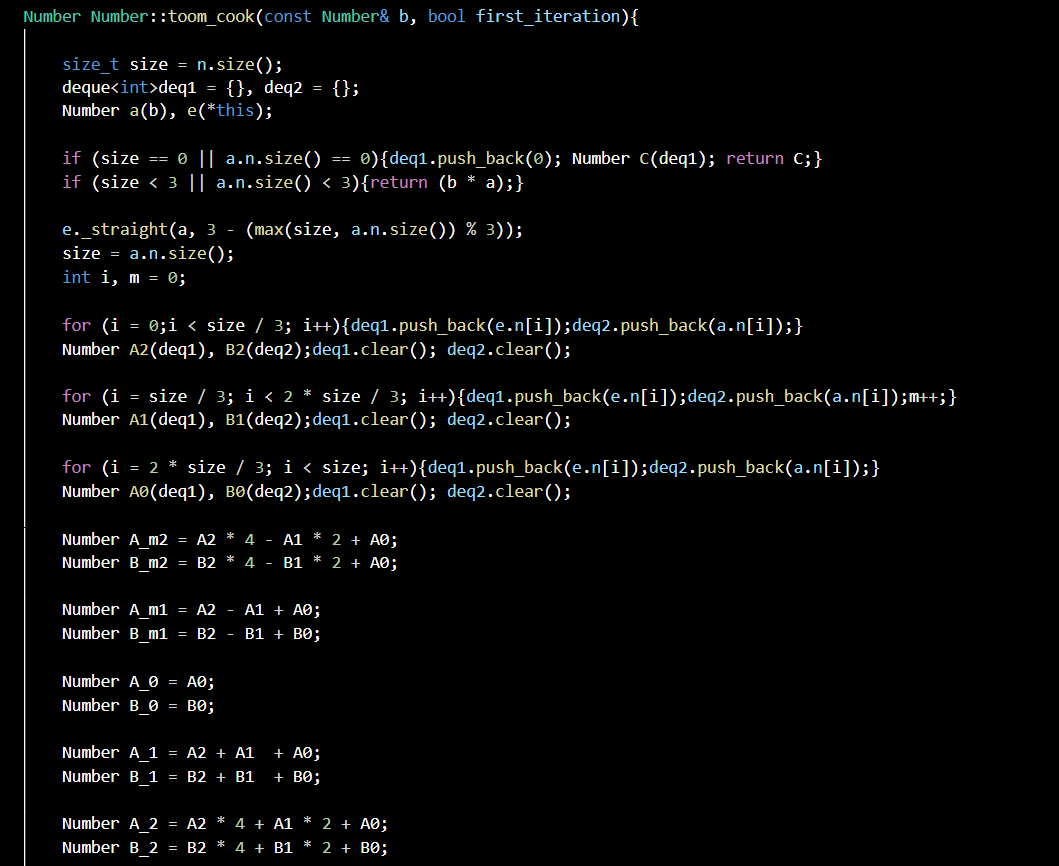
* 1. **Множення Карацуби**

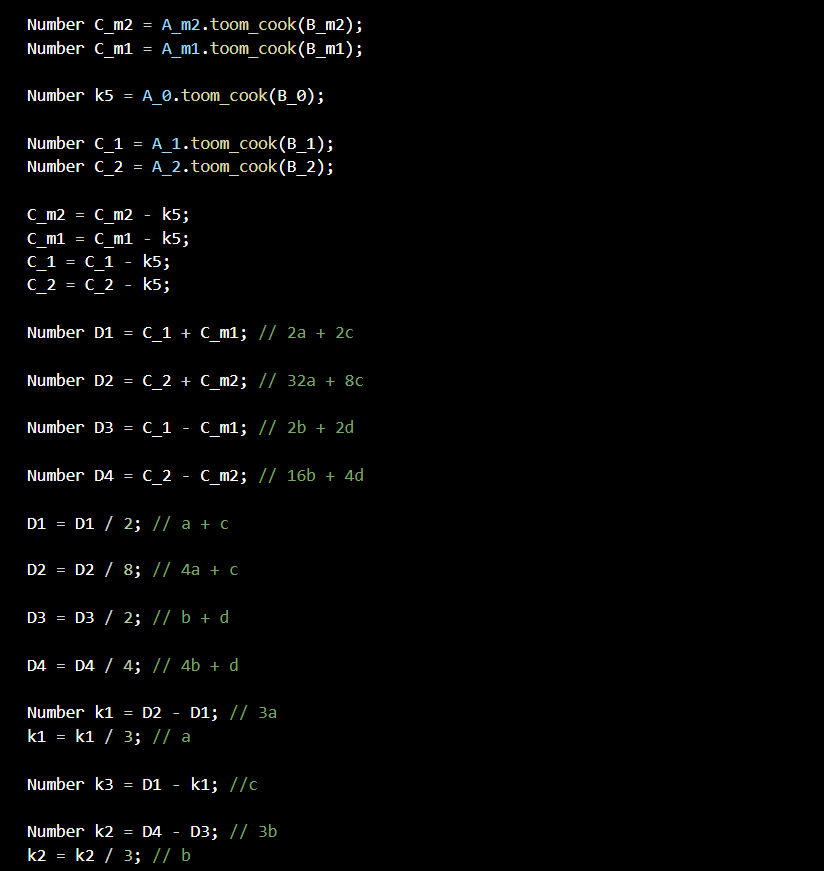
Алгоритм передбачає розділення кожного числа на 2 частини і поочергове перемноження цих частин. За рекурсією добуток зводиться до суми добутків з таблиці множення.

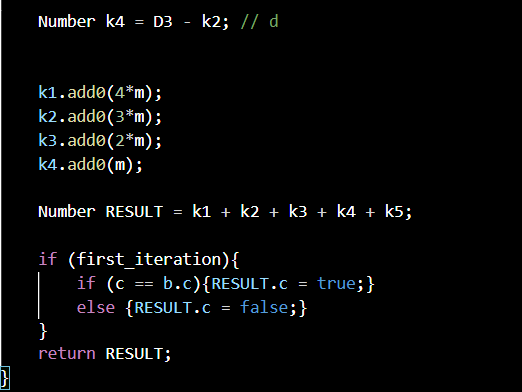


**1.2 Множення Тома Кука**

Алгоритм передбачає розділення кожного числа на 3 частини і зведення добутку до множення двох поліномів. Множення двох поліномів за алгоритмом зводиться до побудови поліному добутку за допомогою знаходження його значень у п’яти точках.

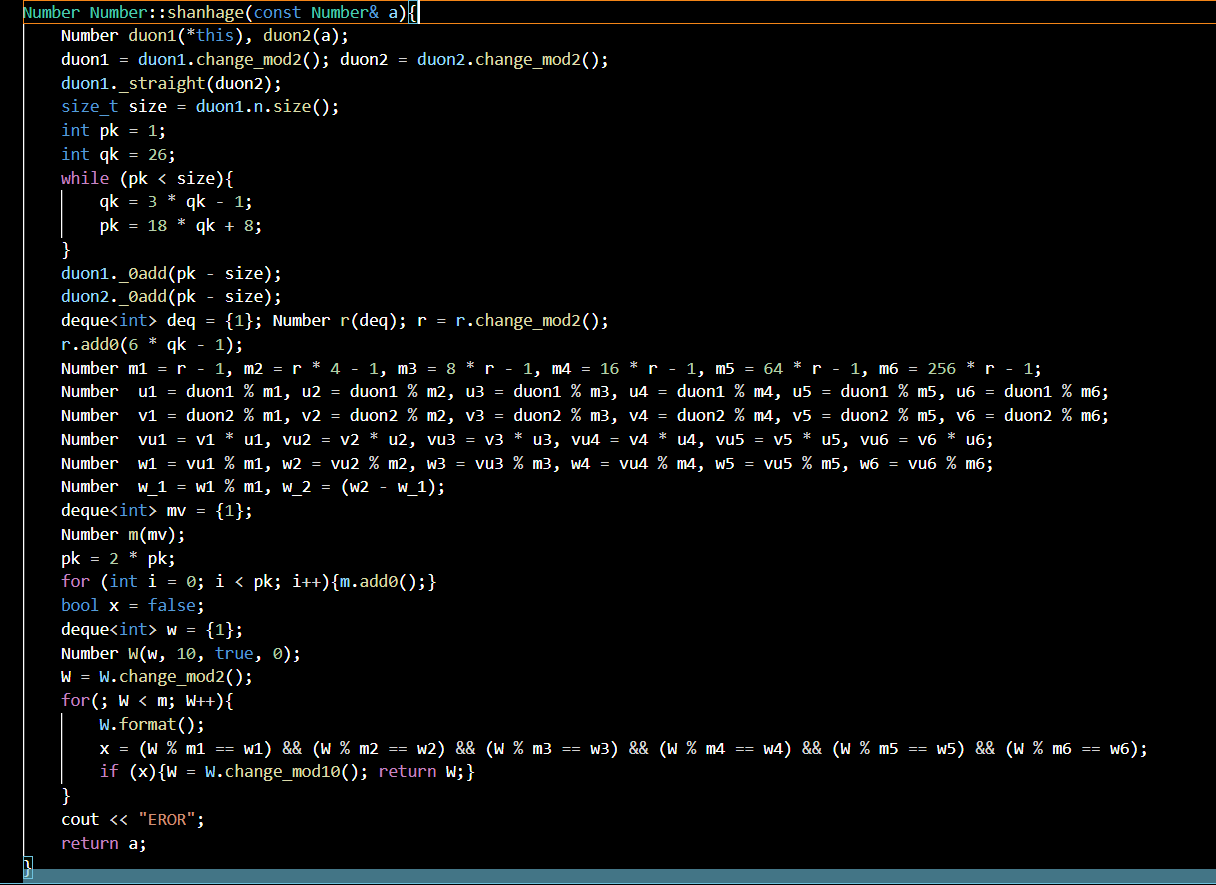






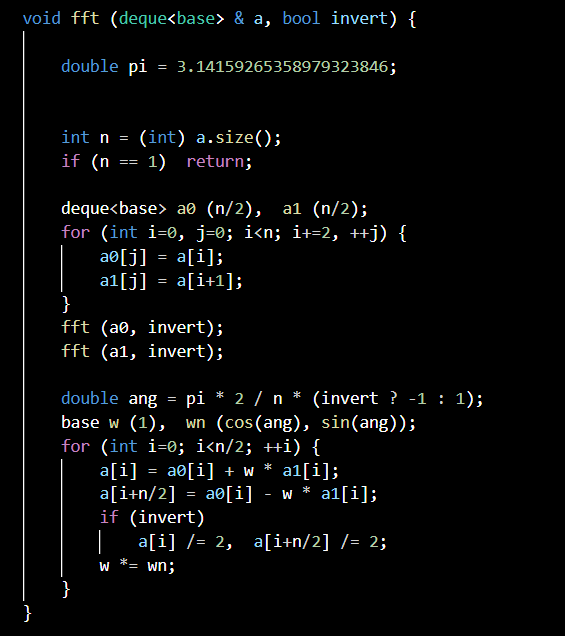
**1.3 Множення за Шанхаге**

Модулярний модуль Шанхаге зводить множення виявився найдовшим ‘’ на короткій дистанції’’ через підбор значення, а не реальне множення. Він підбирає значення добутку за модулями по деяким (простим відносно один одного) числам.

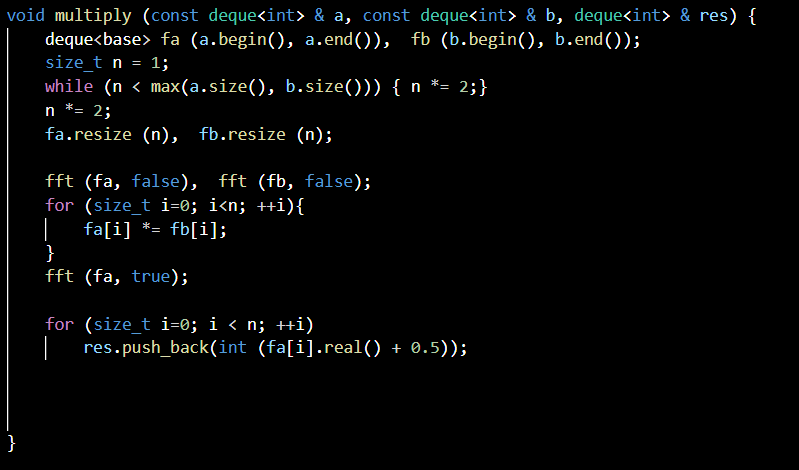


**1.4 Множення Штрассена**

Алгоритм так само, як і Том Кук, розділяє числа на поліноми, але множить їх за допомогою швидкого перетворення Фур’є. Навіть сама природа алгоритмів досить схожа. Том Кук по суті частковий випадок Штрассена.

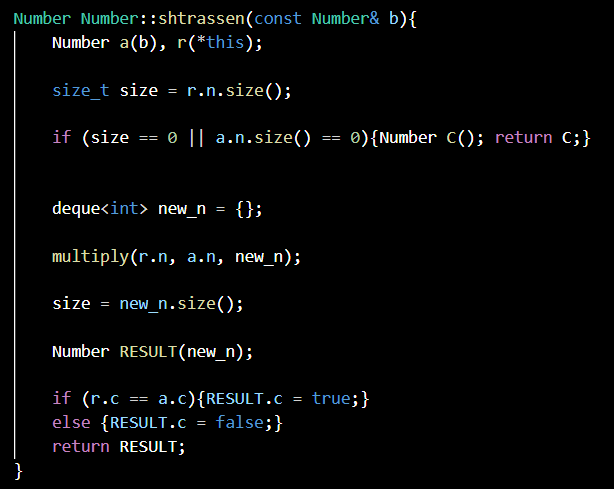
Швидке перетворення Фур’є : 

Множення поліномів за допомогою Ш.П.Ф.



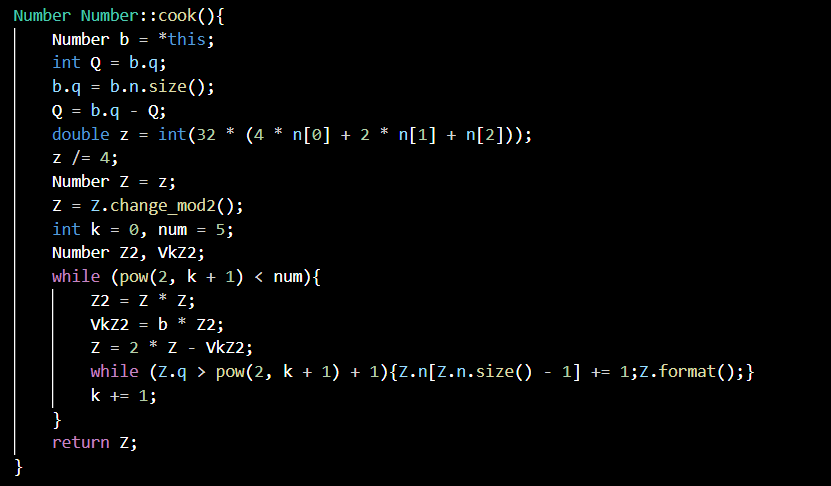
(Для Ш.П.Ф. потрібні комплексні числа з #include<complex>)

Сам алгоритм Штрассена



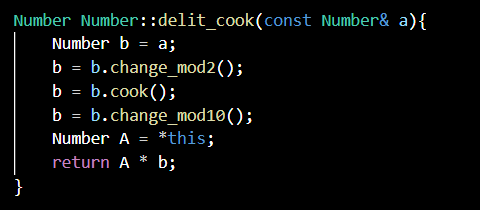
**1.5 Обернена величина за Куком**

Єдина складність алгоритму це робота з дробовими двійковими числами, сам алгоритм досить простий і інтуітивний.



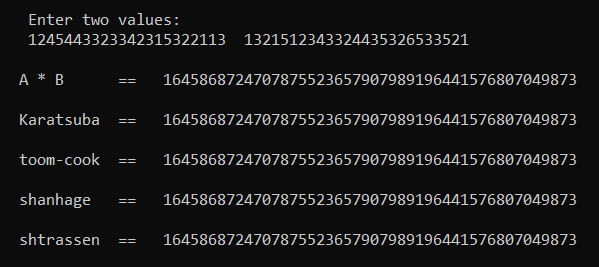
**1.6 Ділення за Куком**

Весь алгоритм це множення діленого на обернене за Куком від дільника.

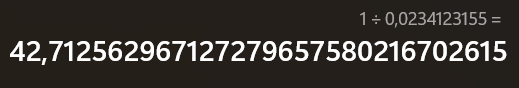
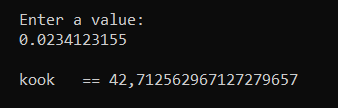


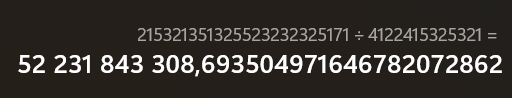
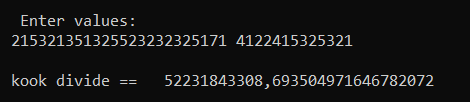
**1.7 Результати множення**

1. Для перевірки множення був написаний оператор множення (у стовпчик)



2. Для перевірки оберненого числа та ділення був використаний гугл калькулятор.

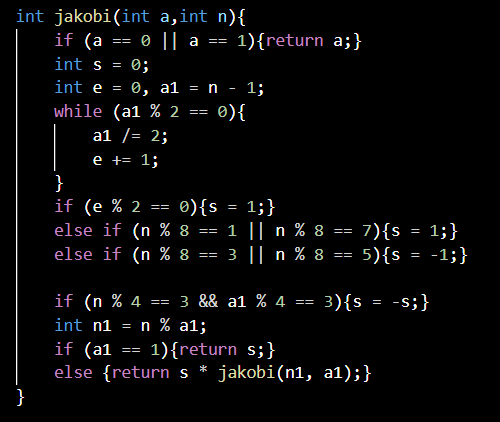




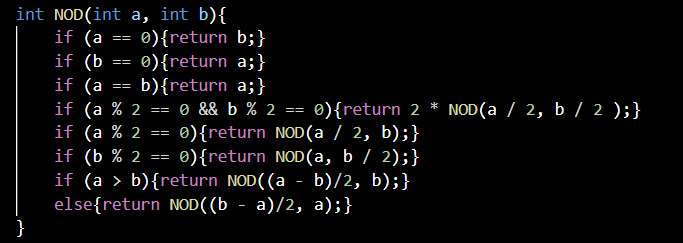
**2 Тести на простоту**

**2.0 Допоміжні функції**

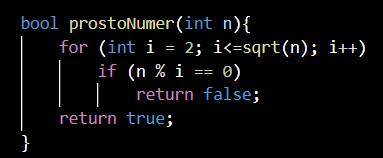
Функція підрахунку символа Якобі



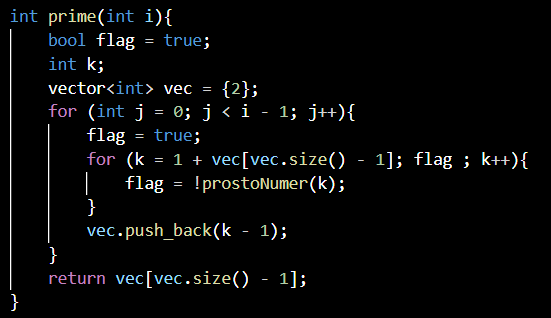
Формула підрахунку ССД



Банальна перевірка на простоту

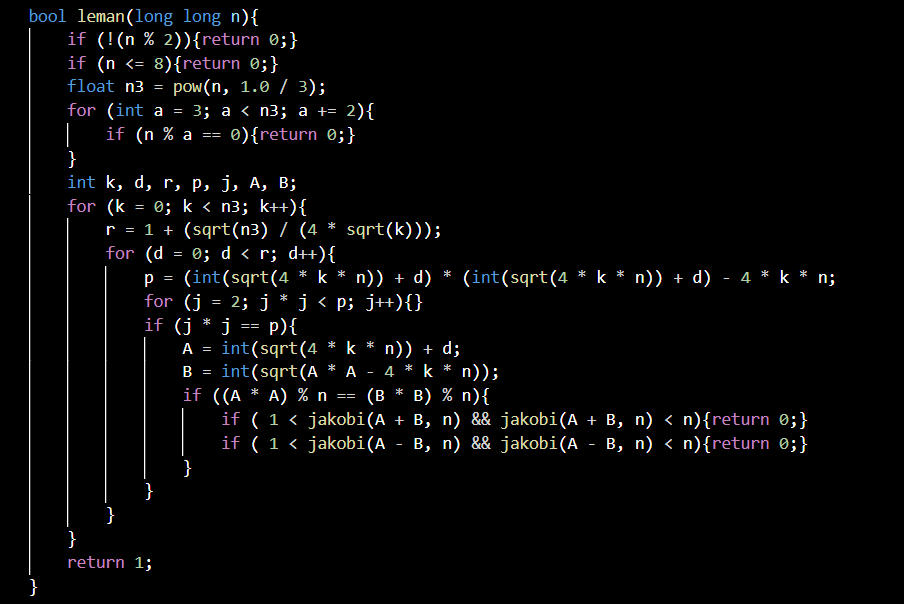


Функція видачі простих чисел



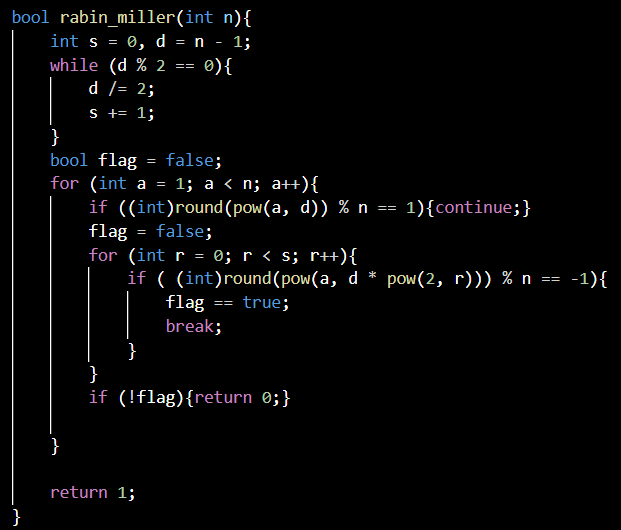
**2.1 Тест Лемана**

Тест оснований на спробах підтвердити составну природу числа n, за допомогою символа Якобі.



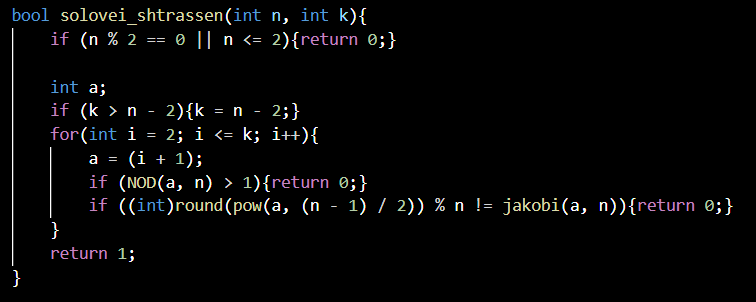
**2.2 Тест Рабіна-Міллера**

Досить ненадійний тест, проте досить популярний через свою простоту.



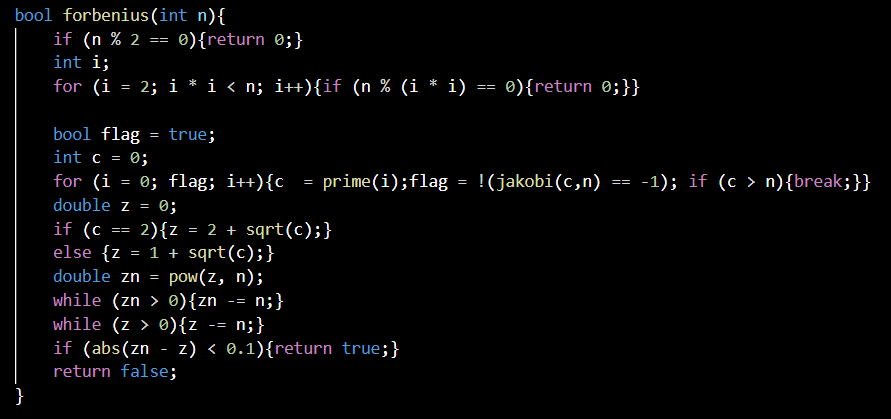
**2.3 Тест Соловей-Штрассена**

Надійність тесту залежить від параметру к , який відповідає за кількість ітерацій циклу перевірок.( зі збільшенням к можна досягти досить високої точності, у джерелах рекомендують використати тест у парі х Рабіном-Міллером)



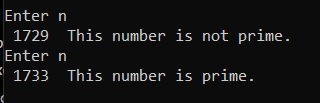
**2.4 Тест Форбеніуса**

Тест оснований на числах Форбеніуса та деяких їх властивостях. Поміж усіх тестів потребує більше теоретичних знань та більш складних обчислень, проте їх він компенсує відносно невеликою кількістю таких обчислень. Досить точний тест.

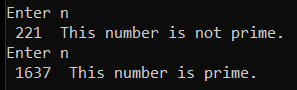


**Результати тестів**

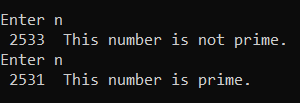
1) Леман



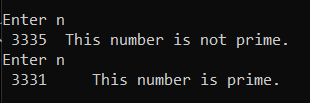
2)Рабін-Міллер не рідко видає лже-прості числа



3)Соловей Штрассен



4) Форбеніус



**Висновки**

* Методи множення довгих чисел можуть з’економити немало часу роботи комп’ютера при множинні досить довгих чисел
* Реалізація більш-менш швидкого ділення довгих чисел потребує знаходження оберненого числа
* Точність і швидкість тесту на простоту частіше за все пропорційна складністю тесту
* Розглянуті у даній лабораторній тести на простоту уже давно перестали бути актуальними через їх ненадійність та повільну роботу
* В 2002 році математики вивели новий тест, який спрацьовує в 100 відсотках випадках і його швидкість збільшують з кожним роком, тож сучасна криптографія не стоїть на місці.