Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій (ІСТ)

Звіт до комп’ютерного практикуму №8

З дисципліни

«Алгоритми та математичні методи захисту інформації»

Підготував:

Студент групи ІК-02

Талько Ярослав

Перевірив:

Нестерук А.

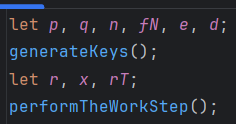
Київ 2024

**Тема:** Протоколи автентифікації.

**Мета:** Ознайомитися з методами простої і строгої автентифікації користувача, практично реалізувати протоколи автентифікації.

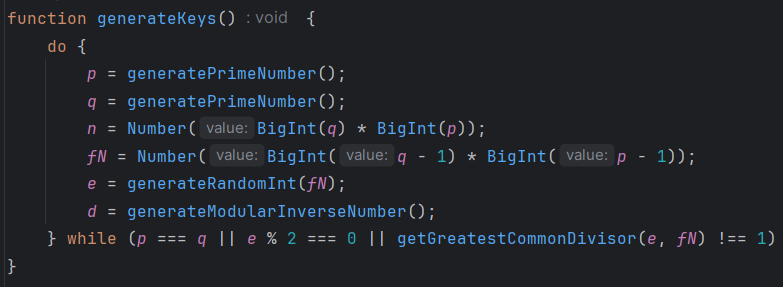
**Хід роботи першого завдання**

Код програми написаний мовою JS. Спочатку розглянемо основну роботу програми в 1 завданні:

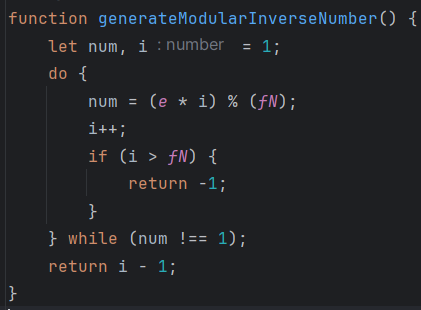


* p та q – два випадкові прості числа;
* n – добуток p та q;
* fN – значення функції Ейлера;
* e – випадкове просто число, взаємно просте з fN і 0<e<fN;
* d – обернене за модулем fN число e;
* r – випадкове число в інтервалі [1, n-1];
* x та rT – змінні, необхідні для перевірки протоколу.

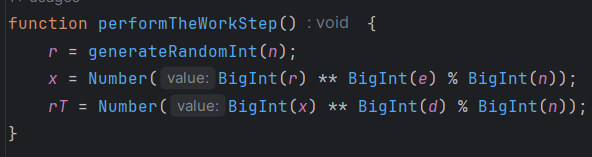
Далі розглянемо функції програми:



Дана функція виконує роль попереднього етапу протоколу.

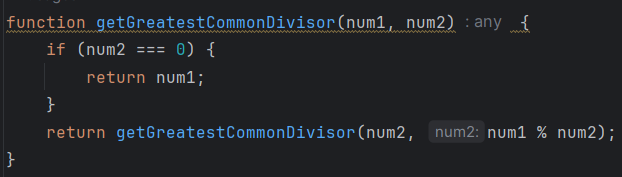


Дана функція займається створенням оберненого по модулю числа.

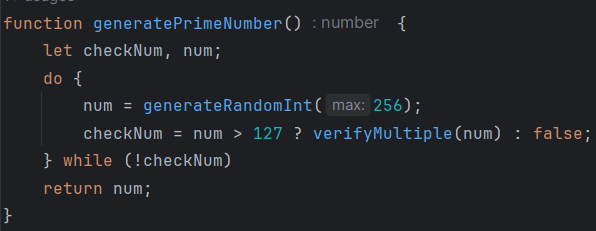


Дана функція виконує робочу частину програми.

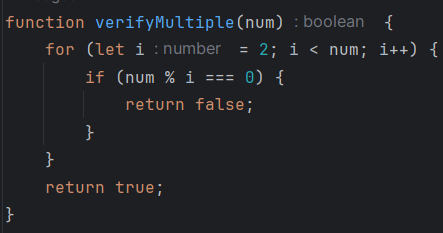
Далі розберемо функції, що зустрічають в обох протоколах:



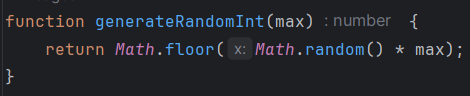
Дана функція повертає НСД двох чисел.



Дана функція займається генеруванням простого числа. Я вирішив виставити його інтервал в межах [128; 255];

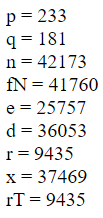
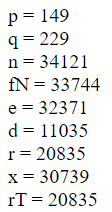
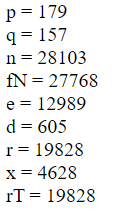


Дана функція перевіряє, чи число справді просте.



Дана функція займається створенням випадкового числа, де max – не включене кінцеве значення;

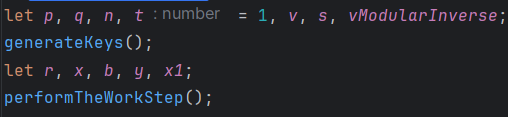
**Результати роботи першої програми:**

****

Одразу представляю кілька скриншотів, аби запевнити, що при різних випадкових значеннях автентифікація успішно підтверджується (r = rT).

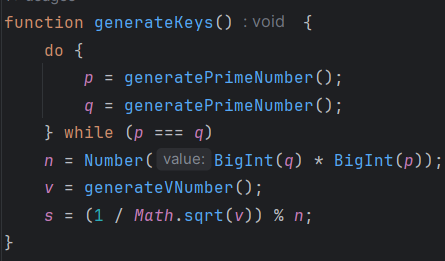
**Хід роботи другого завдання**

Код програми написаний мовою JS. Розглянемо основну роботу програми в другому завданні:

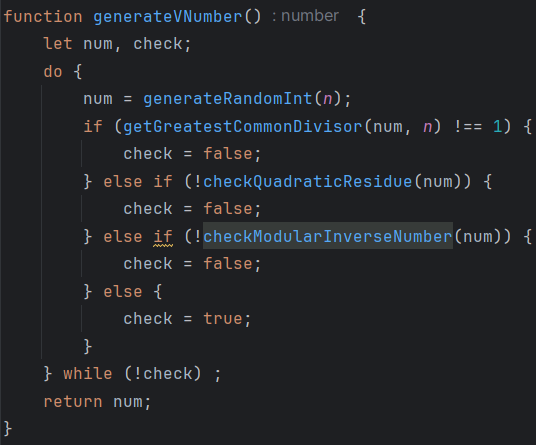


* p та q – секретні нерівні прості числа;
* n – добуток p та q;
* t – число циклів протоколу;
* v – випадкове число, взаємно просте з n, є квадратичним лишком за модулем n та має обернене значення за модулем n;
* s – закритий ключ;
* vModularInverse – обернене за модулем n число v;
* r – випадкове число, взаємно просте з n;
* x – доказ;
* b – випадковий біт;
* y та x1 – змінні, необхідні для перевірки роботи протоколу.

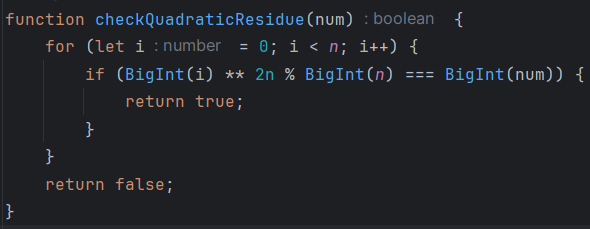
Розглянемо детальніше функції програми:



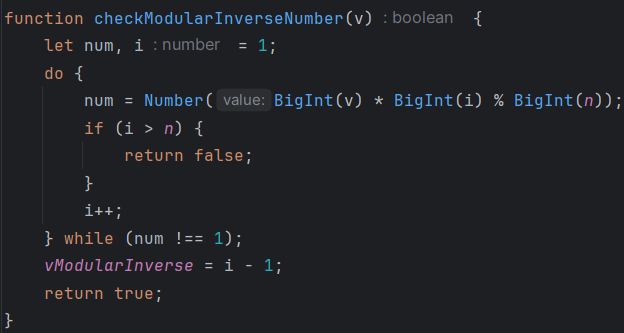
Дана функція займається створенням ключів, та інших змінних згідно з попереднім етапом протоколу.



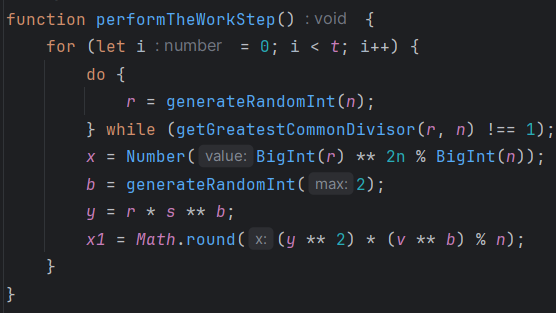
Дана функція створює число v згідно умов алгоритму.



Дана функція перевіряє число на квадратичний лишок.

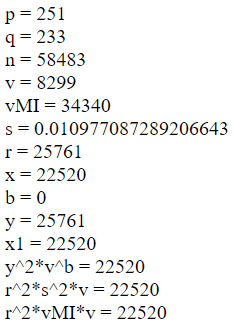
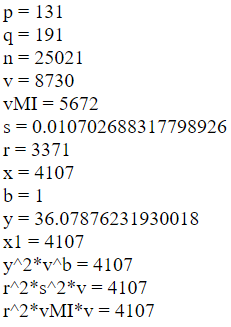
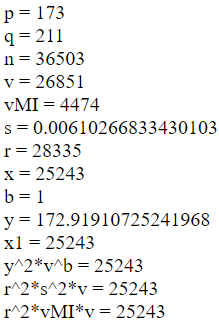


Дана функція перевіряє, чи має число обернене число по модулю.



Дана функція виконує всю робочу частину алґоритму.

**Результати роботи першої програми:**

**** **** 

Одразу представляю кілька скриншотів, аби запевнити, що при різних випадкових значеннях автентифікація успішно підтверджується (x = x1 = (y^2\*v^b)%n = (r^2\*s^2\*v)%n = (r^2\*vModularInverse\*v)%n).

**Висновок:** На цій лабораторній роботі я ознайомився з методами простої і строгої автентифікації користувача, практично реалізував протоколи автентифікації.