**КРИПТОГРАФІЯ**

**КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4**

Вивчення криптосистеми RSA та алгоритму електронного

підпису; ознайомлення з методами генерації параметрів для

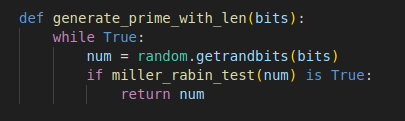
асиметричних криптосистем

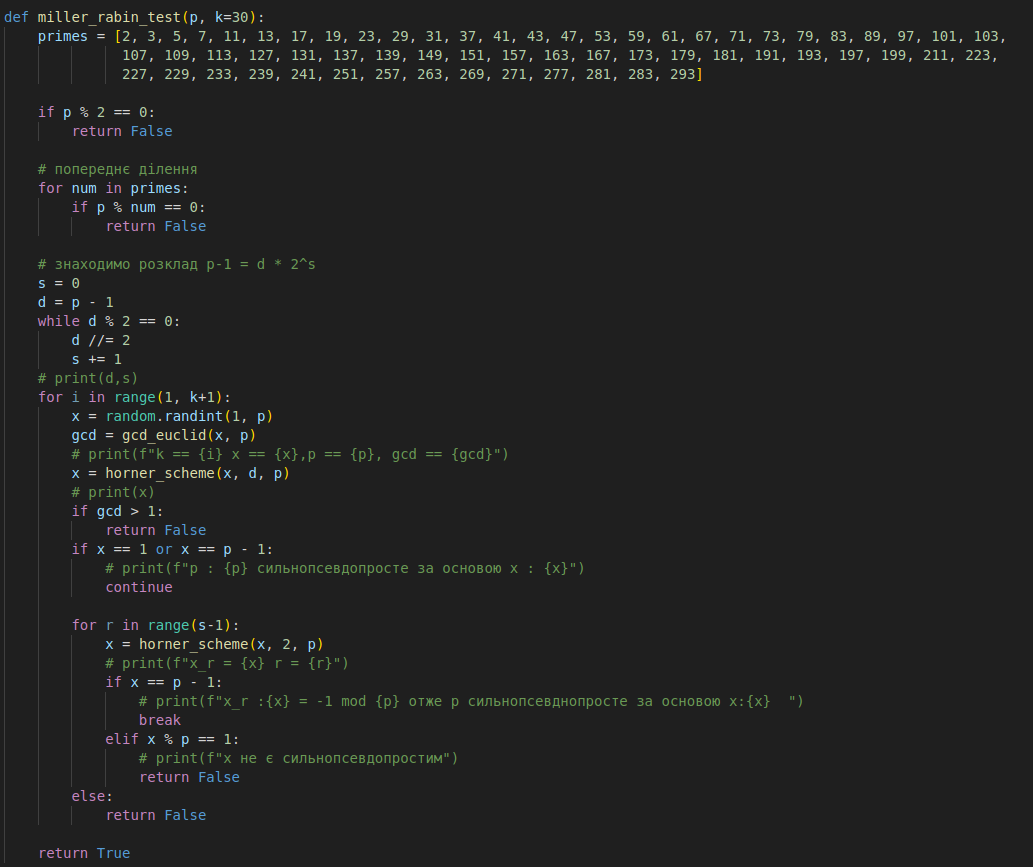
1. Написати функцію пошуку випадкового простого числа з заданого інтервалу або

заданої довжини, використовуючи датчик випадкових чисел та тести перевірки на

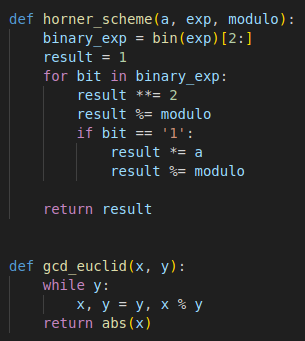
простоту. В якості датчика випадкових чисел використовуйте вбудований генератор

псевдовипадкових чисел вашої мови програмування. В якості тесту перевірки на простоту рекомендовано використовувати тест Міллера-Рабіна із попередніми пробними діленнями. Тести необхідно реалізовувати власноруч, використання готових реалізацій тестів не дозволяється:



Для реалізації цієї функції використаємо тест Міллера Рабіна:

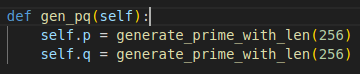
А також функцію схеми Горнера та алгоритм Евкліда:



2. За допомогою цієї функції згенерувати дві пари простих чисел p, q і p1 , q1

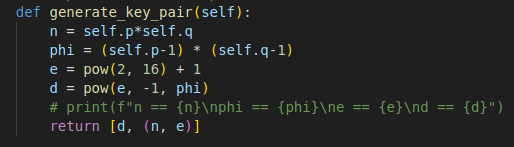
довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq  p1q1 ; p і q – прості числа для побудови ключів абонента А, p1 і q1 – абонента B:

Тепер можемо згенерувати числа p, q:



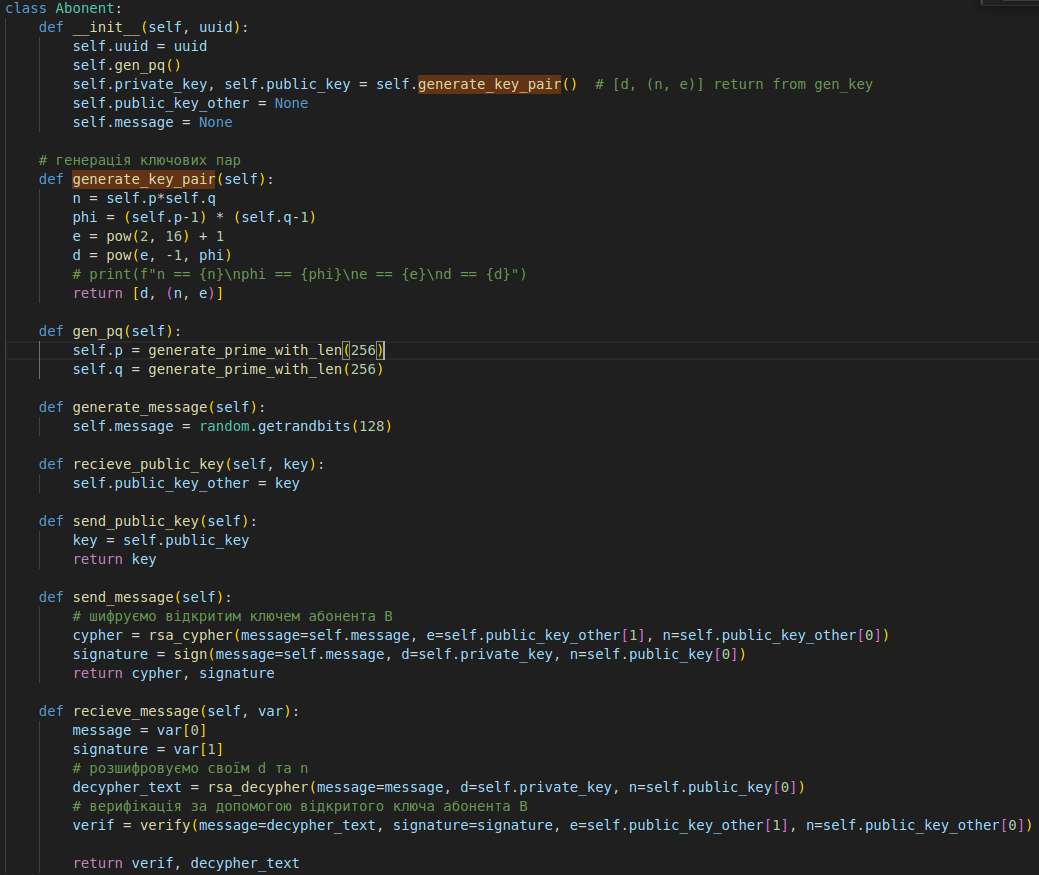
3. Написати функцію генерації ключових пар для RSA. Після генерування функція

повинна повертати та/або зберігати секретний ключ (d , p, q) та відкритий ключ (n, e) . За допомогою цієї функції побудувати схеми RSA для абонентів А і B – тобто, створити та зберегти для подальшого використання відкриті ключі (e, n) , (e1, n1 ) та секретні d і d1 :



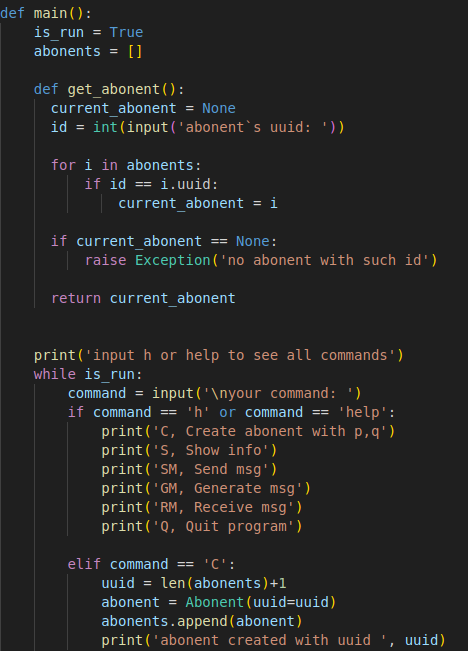
4. Написати програму шифрування, розшифрування і створення повідомлення з

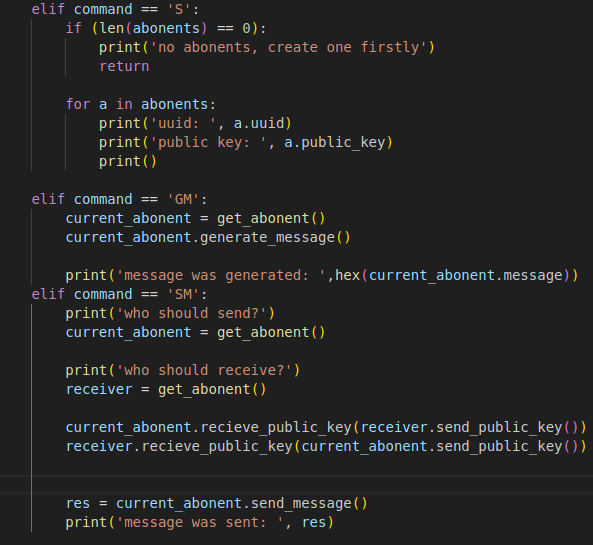
цифровим підписом для абонентів А і B. Кожна з операцій (шифрування, розшифрування, створення цифрового підпису, перевірка цифрового підпису) повинна бути реалізована окремою процедурою, на вхід до якої повинні подаватись лише ті ключові дані, які необхідні для її виконання. За допомогою датчика випадкових чисел вибрати відкрите повідомлення M і знайти криптограму для абонентів А и B, перевірити правильність розшифрування. Скласти для А і B повідомлення з цифровим підписом і перевірити його.

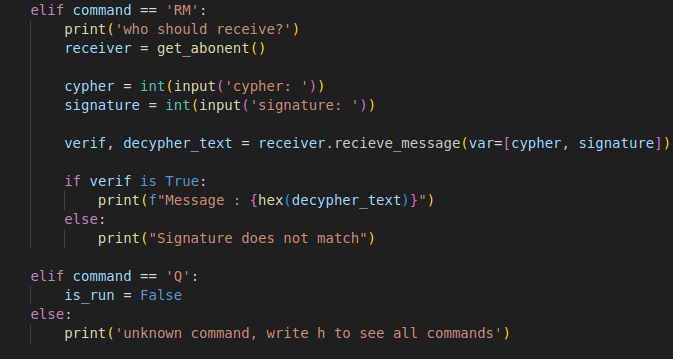


реалізуємо клас який буде в собі містити всі необхідні методи для шифрування та розшифрування повідомлень

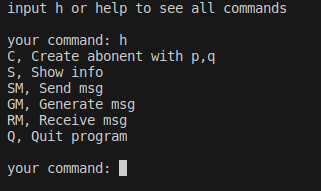
також реалізуємо інтерфейс для взаємодії з цими функціями:



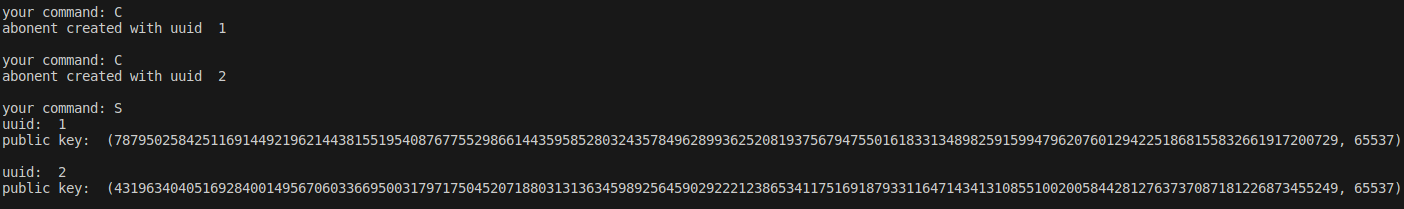




Запустимо нашу програму та перевіримо її функціональність:



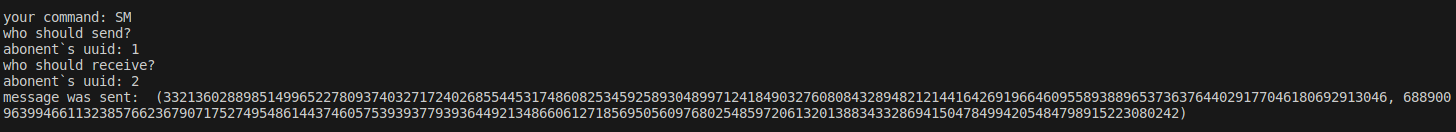
створимо 2-ох користувачів та переглянемо інформацію про них:



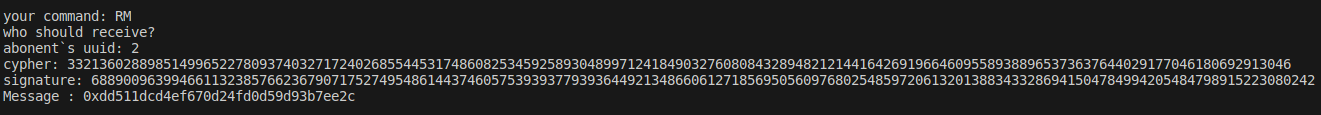
Згенеруємо повідомлення для 1:



Надішлемо повідомлення від 1 користувача до 2:



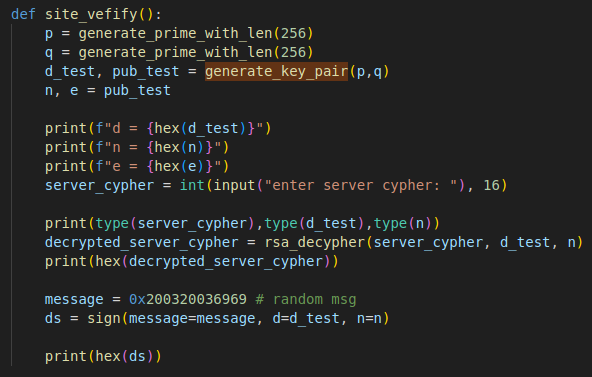
Приймемо та розшифруємо повідомлення 1 користувача яке він надіслав для 2:

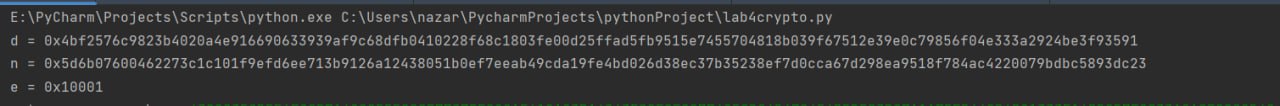


Як бачимо ми отримали те саме повідомлення, яке і надіслали

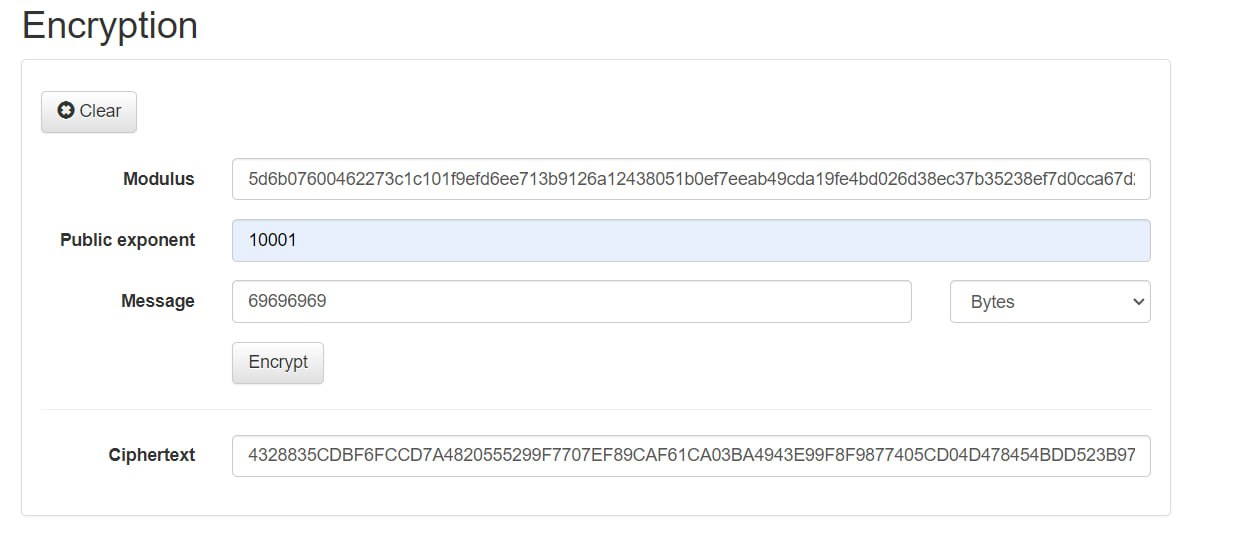
Також перевіримо роботу наших алгоритмів за допомогою стороннього сервісу

для цього напишемо функцію для взаємодії з цим сайтом:

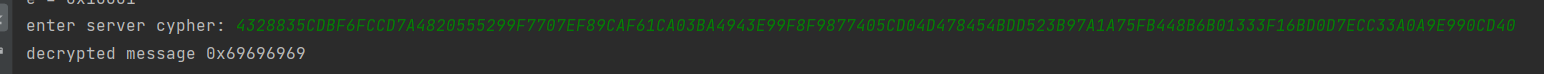


згенеруємо приватний та публічний ключі:

шифруємо на сайті відкритим ключем повідомлення:



розшифровуємо секретним ключем:

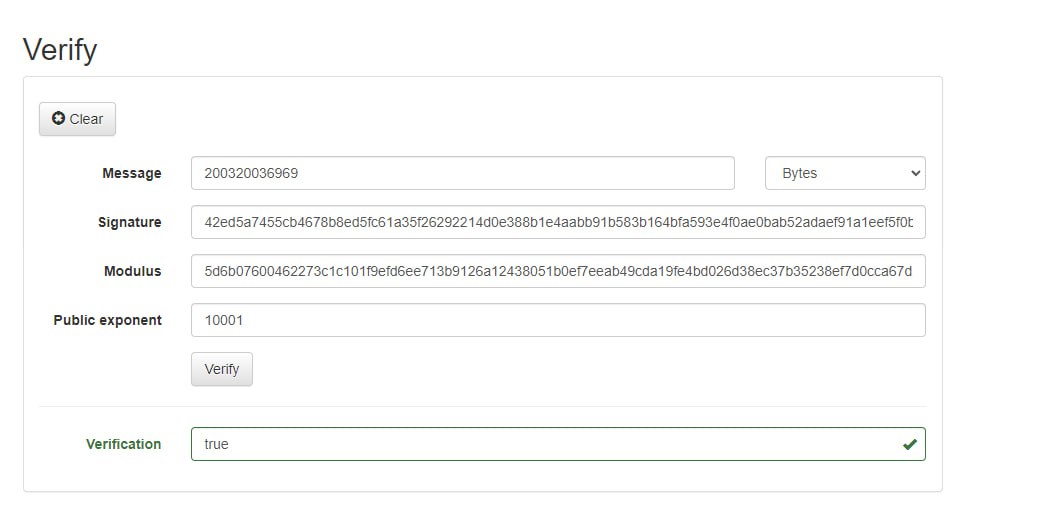


перевіримо цифровий підпис та повідомлення

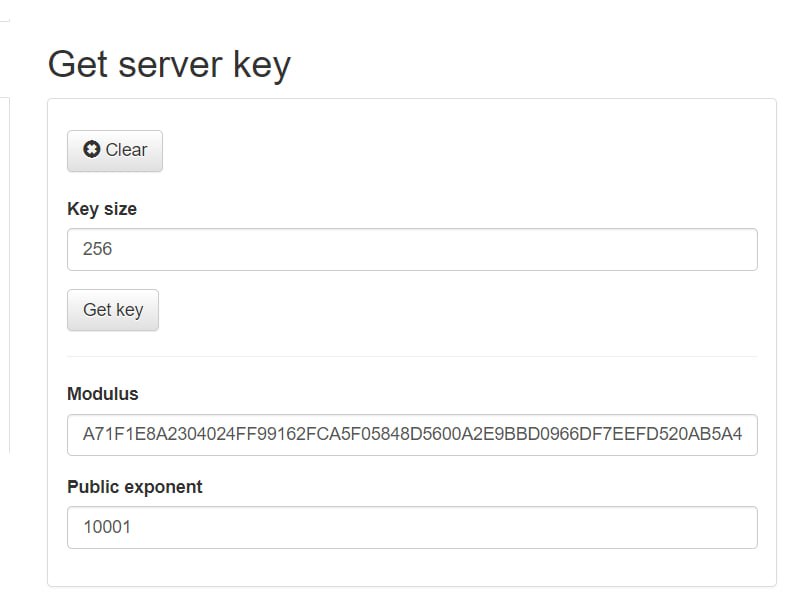




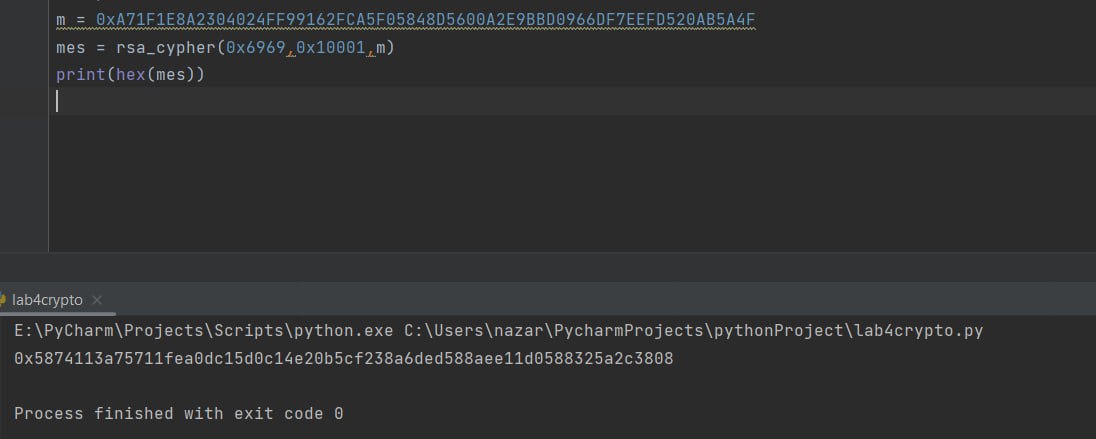
перевіримо сам підпис через сайт:



отримаємо ключ на сайті:



локально зашифруємо на сервері:



розшифруємо на сайті:

