УКРАЇНСЬКИЙ КАТОЛИЦЬКИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНИХ НАУК

Комп’ютерні науки

**Обмін повідомленнями**

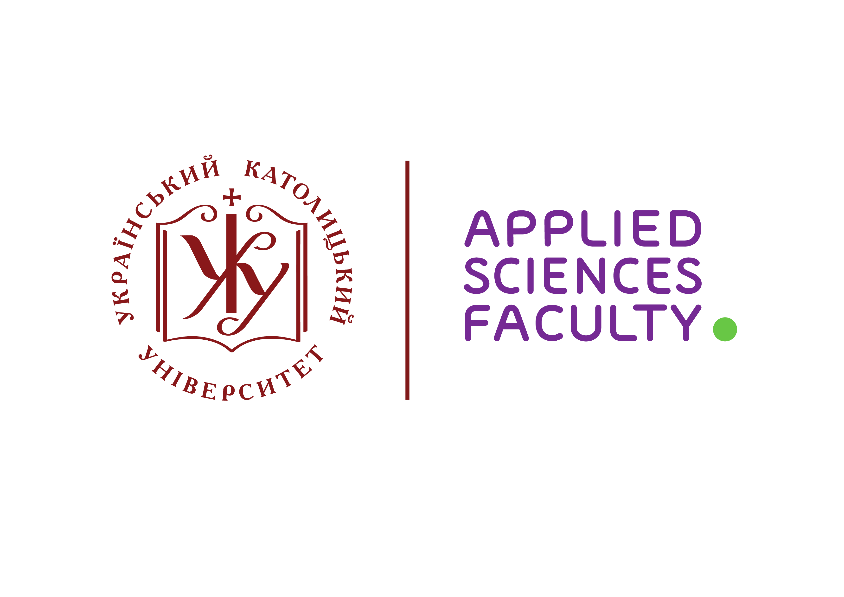
**завдяки шифру RSA**

***Автори:***

*Герасимук Денис та*

*Цапів Володимир*

30 травня 2020



1. **Вступ.**

Наше завдання звучало наступним чином: «Задано англомовне повідомлення, яке закодовано в криптосистемі RSA з відкритим ключем (n, e). Також дано прості числа p і q такі, що n = p\*q. На основі цієї інформації генерувати секретний ключ d і розшифрувати повідомлення.»

Проте ми на цьому не зупинилися та вирішили створити свій власний веб-застосунок, який допоможе не тільки розшифровувати повідомлення, а й зашифровувати їх. Більше того, тепер користувач має змогу працювати з цілими реченнями, бо ми також зашифровуємо всі символи з таблиці ASCII, що включає в себе також пробіли і знаки пунктуації. Детальніше з таблицею можете ознайомитися тут 🡪<http://www.asciitable.com/>

Проект було реалізовано на мові програмування Python. Для шифрування і дешифрування повідомлень використовується простий і розширений алгоритм Евкліда, зокрема для визначення оберненого за модулем числа, і модульне піднесення до степеня, яке неймовірно пришвидчувало процес визначення модуля від числа піднесеного до степеня четвертого порядку. Також було реалізовано кілька функцій для перетворення тексту у формат, з яким працює алгоритм RSA, і назад. Цей проект потребував хороших знань модульної арифметики і розуміння роботи кількох алгоритмів на її основі.

1. **Псевдокод.**

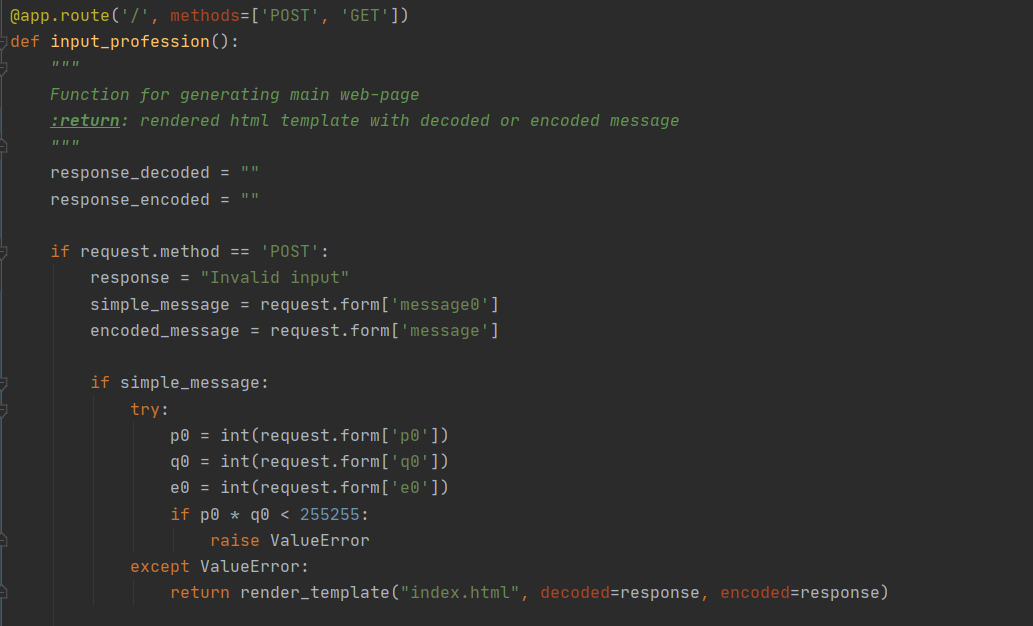
При декодуванні вашого повідомлення потрібно дотримуватися **правила введення**. Так правильним уводом вважається - 149448, 246272, 256490, 197831, 118058, 163098

Але якщо змінити його для прикладу так(добавити зайві пробіли) - 149448, 246272, 256490, 197831, 118058, 163098

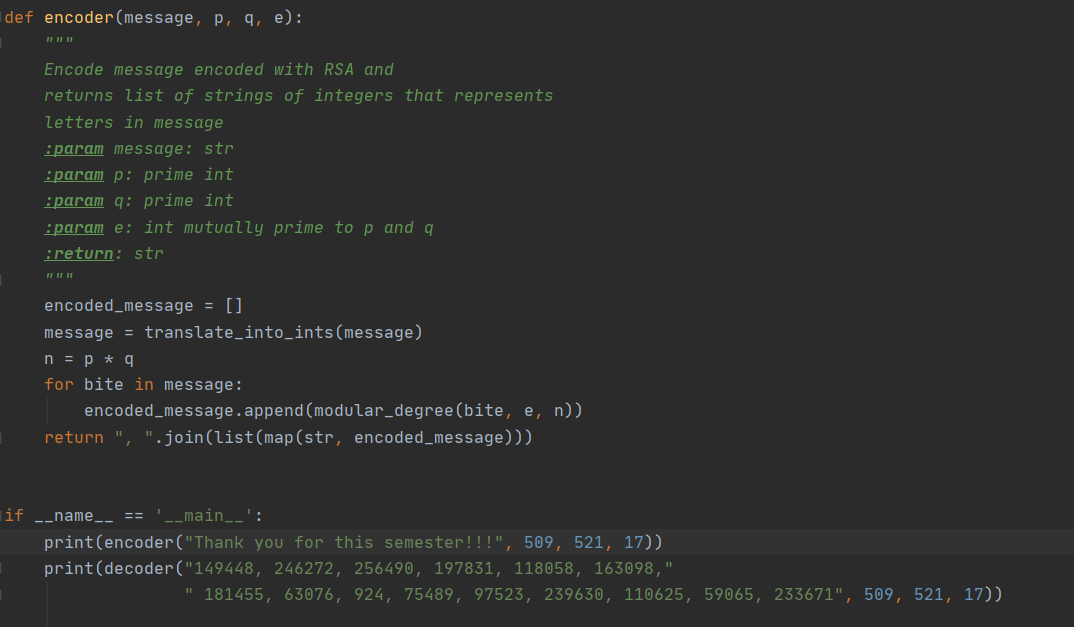
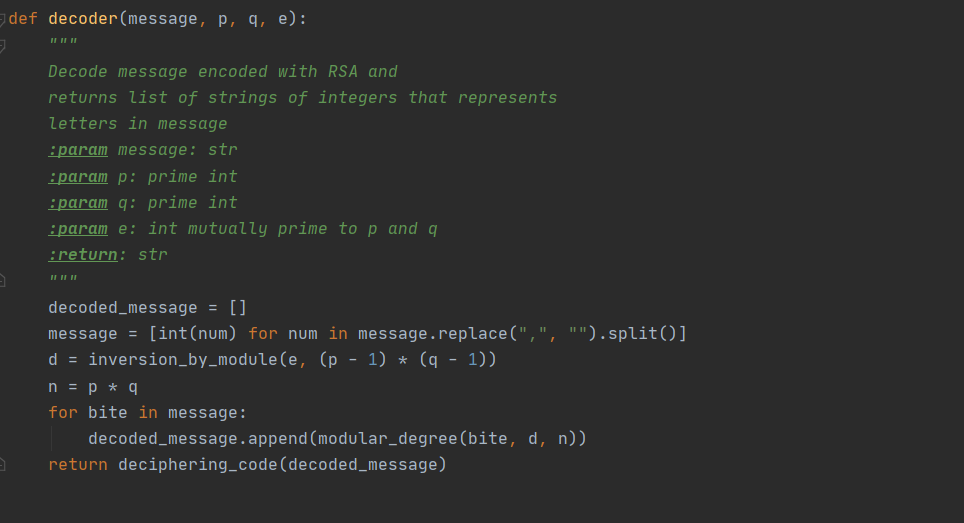
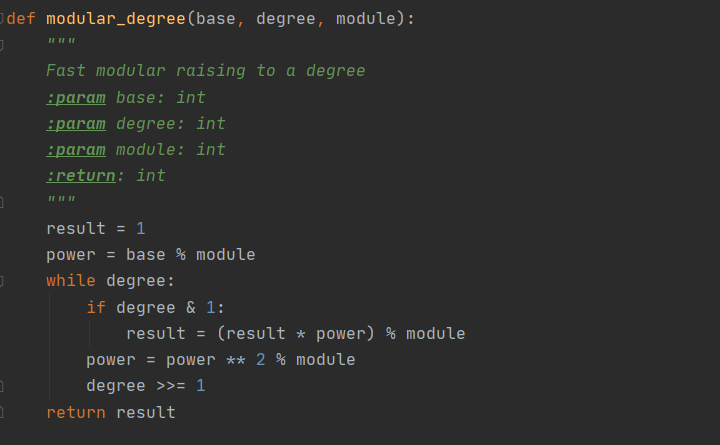
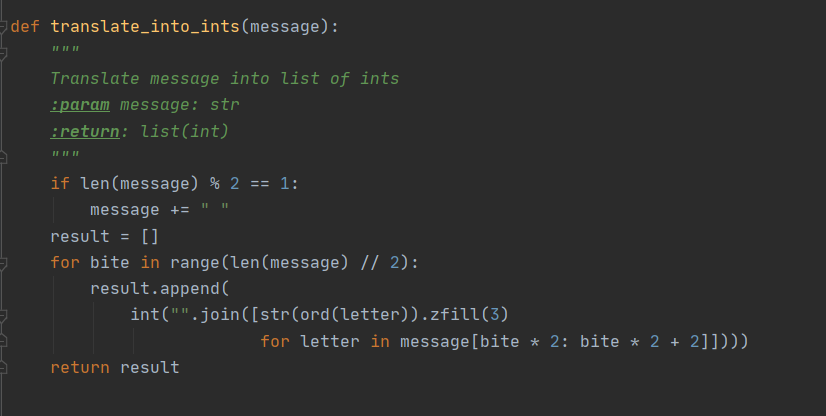
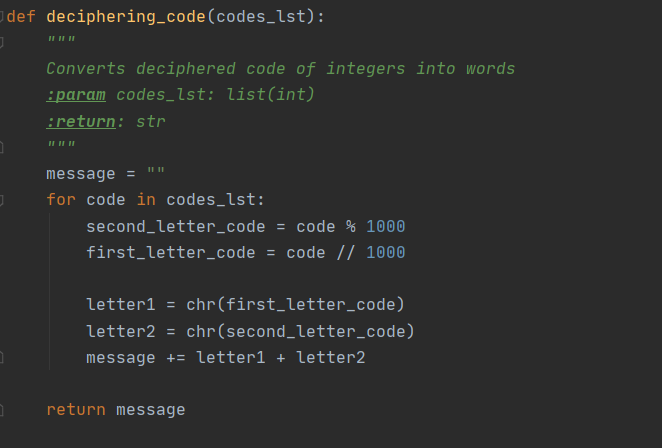
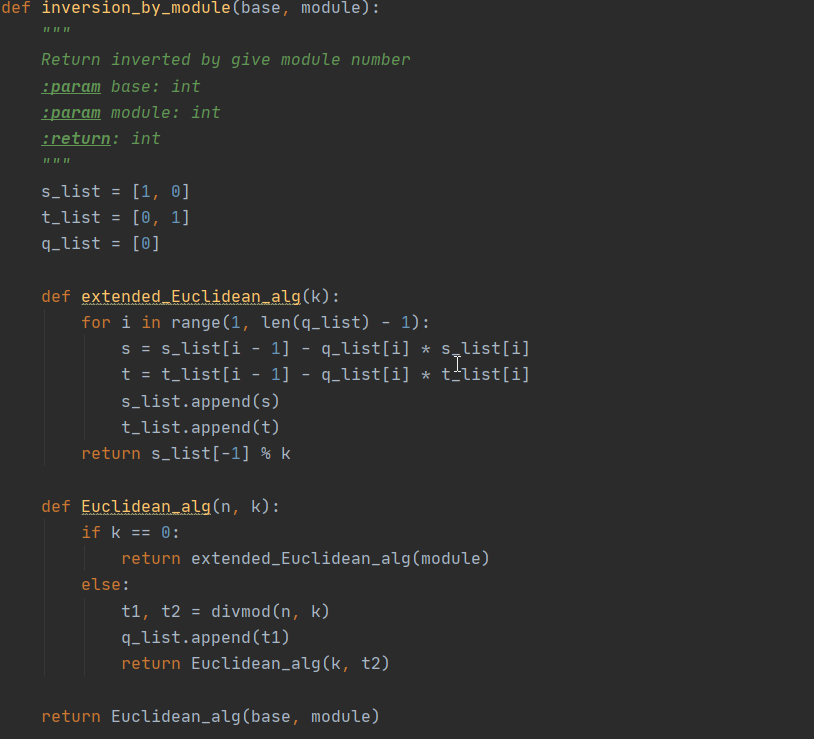
Тоді програма розкодує це, але **вже буде інше повідомлення**, яке відрізнятиметься від того, що було закодоване!!!

Також потрібно вводити для закодовування **текст на англійській мові**, оскільки ми працюємо з ASCII таблицею.

**flask\_app.py**

****

**RSA.py**

****

1. **Висновки.**

На основі даного комп’ютерного проекту ми навчилися використовувати RSA шифр як для шифрування, так і для розшифрування(для перевірки потрібно дві функції). Також нам вдалося практично застосувати знання з курсу дискретної математики і побачити переваги та недоліки даного методу шифрування. Серед них те, що цим шифром важко обмінюватися довгими реченнями із пробілами та знаками пунктуації. Тому ми рекомендуємо використовувати наш сервіс 🡪 <https://rsa-decoder.herokuapp.com/>

**Список плюсів** асиметричного шифрування

1. Це дозволяє аутентифікацію повідомлень.

Оскільки шифрування відкритого ключа дозволяє використовувати цифровий підпис, одержувачі повідомлень зможуть перевірити, чи справді вони надходять від певного відправника.

2. Це зручно.

Асиметричне шифрування вирішує проблему розподілу ключів для шифрування, кожен публікує свої відкриті ключі, а приватні ключі зберігаються в секреті.

3. Це дозволяє неприйняття.

Цифрово підписані повідомлення - це як фізично підписані документи. В основному це схоже на підтвердження повідомлення, а отже, відправник не зможе його відмовити.

4. Він виявляє фальсифікацію.

За допомогою цифрових підписів при шифруванні відкритого ключа, одержувачі повідомлень можуть виявити, чи було змінено повідомлення під час транзиту.

**Список мінусів** асиметричного шифрування

1. Це повільний процес.

Шифрування відкритого ключа в цьому методі повільне порівняно з симетричним шифруванням, а це означає, що воно не підходить для розшифровки масових повідомлень.

2. Його відкриті ключі не засвідчуються автентифікацією.

В основному, ніхто абсолютно не знає, що публічний ключ належить особі, яку він вказує, а це означає, що користувачам доведеться перевірити, чи їхні відкриті ключі дійсно належать їм.

3. Це ризикує втратою приватного ключа, що може бути непоправною.

Коли ви втратите приватний ключ, отримані повідомлення не розшифруються.

4. Це загрожує широким компромісом з безпекою.

Якщо ваш приватний ключ зловмисник ідентифікує, усі ваші повідомлення можуть його читати.(атака по середині)