**Харківський національний економічний університет**

**імені Семена Кузнеця**

**ЗВІТ**

**З ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 8**

**за дисципліною: “Теоретичні основи криптографії”**

**Виконав: студент факультету Інформаційних технологій**

**2 курсу, спец. Кібербезпека,**

**групи 6.04.125.010.21.2**

**Бойко Вадим Віталійович**

**Перевірив:**

**Чугай Андрій Михайлович**

**ХНЕУ ім. С. Кузнеця**

**2023**

**Тема:** Дослідження криптоалгоритму шифрування RSA

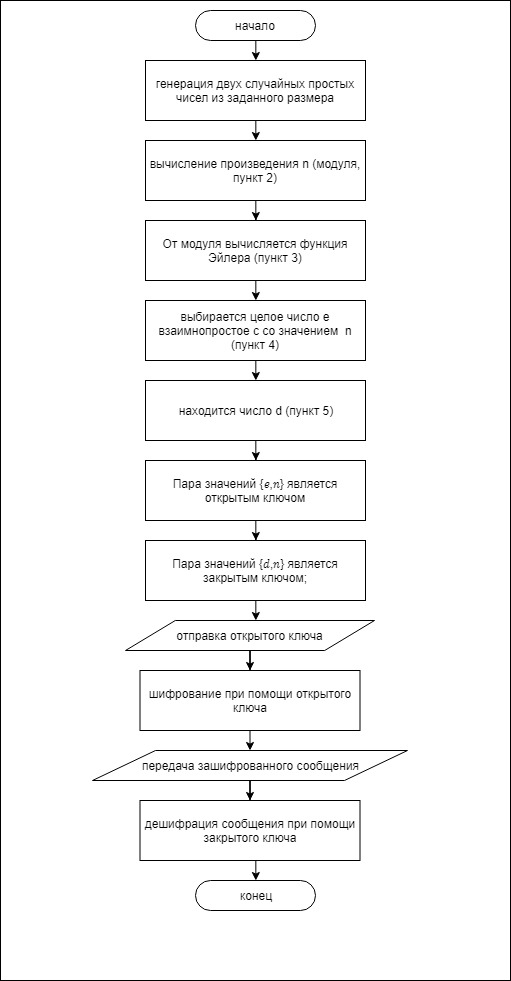
**Мета:** Дослідити структуру алгоритму та методики практичної реалізації криптосистеми шифрування RSA.

**Завдання:**

1. Тему, мету і порядок роботи.
2. Блок-схему та програму алгоритму шифрування RSA.
3. Висновки: переваги й недоліки алгоритму шифрування RSA.

Хід роботи

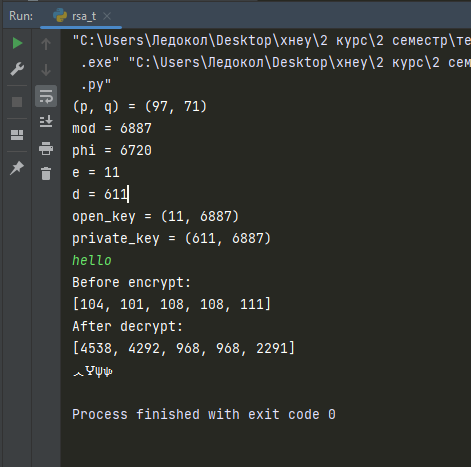
Блок схема:

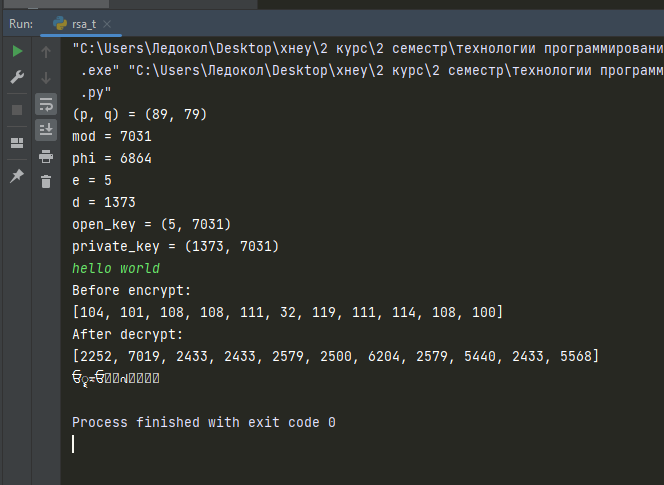


Код:

*from* sympy *import* randprime, primerange  
*from* math *import* gcd  
  
  
*def* find\_two\_primes():  
 p1 = randprime(1, 100)  
 p2 = randprime(1, 100)  
 *while* p1 == p2:  
 p2 = randprime(1, 100)  
 *return* (p1, p2)  
  
  
a = find\_two\_primes()  
print(f'(p, q) = {a}')  
mod = a[0] \* a[1]  
print(f'mod = {mod}')  
phi = (a[0] - 1) \* (a[1] - 1)  
print(f'phi = {phi}')  
spisok = list(primerange(1, phi))  
  
*for* n *in* spisok:  
 *if* gcd(n, phi) == 1:  
 e = n  
 *break*values = [x *for* x *in* range(1, phi + 1)]  
d = 0  
*for* elem *in* values:  
 *if* (int(elem) \* e) % phi == 1:  
 d += int(elem)  
 *break*print(f'e = {e}')  
print(f'd = {d}')  
open\_key = (e, mod)  
private\_key = (d, mod)  
print(f'open\_key = {open\_key}')  
print(f'private\_key = {private\_key}')  
b = list(input())  
b = [ord(elem) *for* elem *in* b]  
print("Before encrypt:")  
print(b)  
*for* i *in* range(len(b)):  
 b[i] \*\*= e  
 b[i] %= mod  
print("After decrypt:")  
print(b)  
print(''.join([chr(el) *for* el *in* b]))

Тест:





Переваги алгоритму:

* Алгоритм є асиметричним, тобто він полягає в поширенні відкритих ключів у мережі. Це дозволяє кільком користувачам обмінюватися інформацією, по незахищеним каналів зв’язку
* користувач сам може змінювати як числа, і відкритий і закритий ключ на власний розсуд, потім він має поширити відкритий ключ у мережі. Це дає можливість збільшити криптостійкість.

Недоліки алгоритму:

* невисока швидкість роботи. Алгоритм RSA працює повільніше симетричного алгоритму DES