**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

по дисциплине **«Методы защиты информации»**

на тему: «**Стандарты симметричного шифрования DES и**

**ГОСТ 28147-89**»

Выполнили: студент гр. ИП-41

Шевкунова В.А.

Коваленко А.И.

Принял: преподаватель

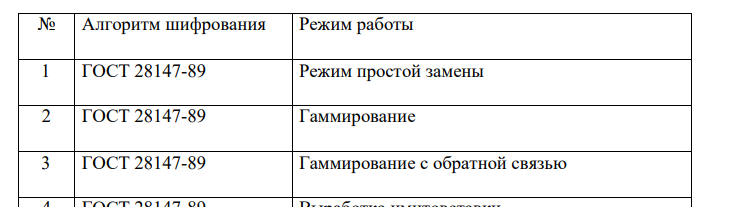
Бородин Н.Н.

Гомель 2023

**Цель**: получить практические навыки решения задач целочисленного программирования.

**Задание** I. Реализовать приложение для шифрования, позволяющее выполнять следующие действия: 1. Шифровать данные по заданному в варианте алгоритму: ‒ шифруемый текст должен храниться в одном файле, а ключ шифрования – в другом; ‒ зашифрованный текст должен сохраняться в файл; ‒ в процессе шифрования предусмотреть возможность просмотра и изменения ключа, шифруемого и зашифрованного текстов в шестнадцатеричном и символьном виде; II. Реализовать приложение для дешифрования, позволяющее выполнять следующие действия: 2. Дешифровать данные по заданному в варианте алгоритму: ‒ зашифрованный текст должен храниться в одном файле, ключ – в другом; ‒ расшифрованный текст должен сохраняться в файл; ‒ в процессе дешифрования предусмотреть возможность просмотра и изменения ключа, зашифрованного и расшифрованного текстов в шестнадцатеричном и символьном виде.

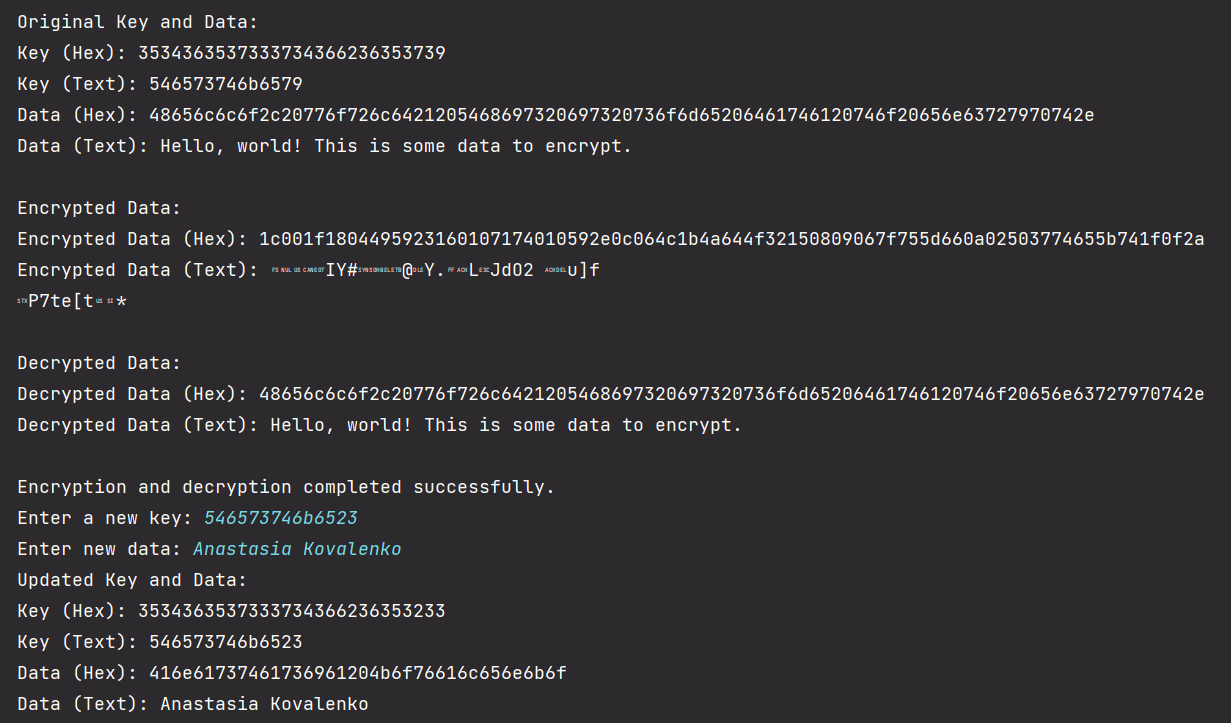
Вариант 3



**Листинг программы:**

import binascii  
  
  
def gost\_encrypt(*key*, *data*):  
 *# Преобразование ключа в список байтов* key\_bytes = *bytearray*.fromhex(*key*)  
  
 *# Преобразование данных в список байтов* data\_bytes = *bytearray*(*data*.encode())  
  
 *# Гаммирование с обратной связью* iv = *bytearray*(8) *# Инициализирующий вектор* encrypted\_data = *bytearray*()  
  
 for i in *range*(*len*(data\_bytes)):  
 *# Шифрование очередного байта данных* encrypted\_byte = data\_bytes[i] ^ iv[i % 8] ^ key\_bytes[i % *len*(key\_bytes)]  
 encrypted\_data.append(encrypted\_byte)  
  
 *# Обновление инициализирующего вектора* iv[i % 8] = encrypted\_byte  
  
 *# Возвращение зашифрованных данных в виде списка байтов* return encrypted\_data  
  
  
def gost\_decrypt(*key*, *data*):  
 *# Преобразование ключа в список байтов* key\_bytes = *bytearray*.fromhex(*key*)  
  
 *# Гаммирование с обратной связью* iv = *bytearray*(8) *# Инициализирующий вектор* decrypted\_data = *bytearray*()  
  
 for i in *range*(*len*(*data*)):  
 *# Расшифровка очередного байта данных* decrypted\_byte = *data*[i] ^ iv[i % 8] ^ key\_bytes[i % *len*(key\_bytes)]  
 decrypted\_data.append(decrypted\_byte)  
  
 *# Обновление инициализирующего вектора* iv[i % 8] = *data*[i]  
  
 *# Возвращение расшифрованных данных в виде строки* return decrypted\_data.decode()  
  
def save\_bytes\_to\_file(*filename*, *data*):  
 with *open*(*filename*, "wb") as f:  
 f.write(data)  
  
  
def save\_string\_to\_file(filename, data):  
 with open(filename, "w") as f:  
 f.write(data)  
  
  
def read\_bytes\_from\_file(filename):  
 with open(filename, "rb") as f:  
 return f.read()  
  
  
def read\_string\_from\_file(filename):  
 with open(filename, "r") as f:  
 return f.read()  
  
  
def print\_hex\_and\_text(data, label):  
 hex\_data = binascii.hexlify(data).decode()  
 text\_data = data.decode()  
 print(f"{label} (Hex): {hex\_data}")  
 print(f"{label} (Text): {text\_data}")  
  
  
def main():  
 key\_file = "key.txt"  
 data\_file = "data.txt"  
 encrypted\_file = "encrypted.txt"  
 decrypted\_file = "decrypted.txt"  
  
 *# Чтение ключа из файла* key = read\_string\_from\_file(key\_file)  
  
 *# Чтение данных из файла* data = read\_string\_from\_file(data\_file)  
  
 *# Вывод исходных данных* print("Original Key and Data:")  
 print\_hex\_and\_text(key.encode(), "Key")  
 print\_hex\_and\_text(data.encode(), "Data")  
 print()  
  
 *# Шифрование данных* encrypted\_data = gost\_encrypt(key, data)  
  
 *# Вывод зашифрованных данных* print("Encrypted Data:")  
 print\_hex\_and\_text(encrypted\_data, "Encrypted Data")  
 print()  
  
 *# Сохранение зашифрованных данных в файл* save\_bytes\_to\_file(encrypted\_file, encrypted\_data)  
  
 *# Загрузка зашифрованных данных из файла* encrypted\_data = read\_bytes\_from\_file(encrypted\_file)  
  
 *# Дешифрование данных* decrypted\_data = gost\_decrypt(key, encrypted\_data)  
  
 *# Вывод расшифрованных данных* print("Decrypted Data:")  
 print\_hex\_and\_text(decrypted\_data.encode(), "Decrypted Data")  
 print()  
  
 *# Сохранение расшифрованных данных в файл* save\_string\_to\_file(decrypted\_file, decrypted\_data)  
  
 *# Вывод успешного завершения операции* print("Encryption and decryption completed successfully.")  
  
 *# Изменение ключа* new\_key = input("Enter a new key: ")  
 key = new\_key.strip()  
  
 *# Изменение данных* new\_data = input("Enter new data: ")  
 data = new\_data.strip()  
  
 *# Вывод обновленных данных* print("Updated Key and Data:")  
 print\_hex\_and\_text(key.encode(), "Key")  
 print\_hex\_and\_text(data.encode(), "Data")  
 print()  
  
 *# Шифрование обновленных данных* encrypted\_data = gost\_encrypt(key, data)  
  
 *# Вывод зашифрованных данных* print("Updated Encrypted Data:")  
 print\_hex\_and\_text(encrypted\_data, "Encrypted Data")  
 print()  
  
 *# Сохранение обновленных зашифрованных данных в файл* save\_bytes\_to\_file(encrypted\_file, encrypted\_data)  
  
 *# Загрузка обновленных зашифрованных данных из файла* encrypted\_data = read\_bytes\_from\_file(encrypted\_file)  
  
 *# Дешифрование обновленных данных* decrypted\_data = gost\_decrypt(key, encrypted\_data)  
  
 *# Вывод расшифрованных обновленных данных* print("Updated Decrypted Data:")  
 print\_hex\_and\_text(decrypted\_data.encode(), "Decrypted Data")  
 print()  
  
 *# Сохранение расшифрованных обновленных данных в файл* save\_string\_to\_file(decrypted\_file, decrypted\_data)  
  
 *# Вывод успешного завершения операции* print("Encryption and decryption of updated data completed successfully.")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Результат работы.**

****

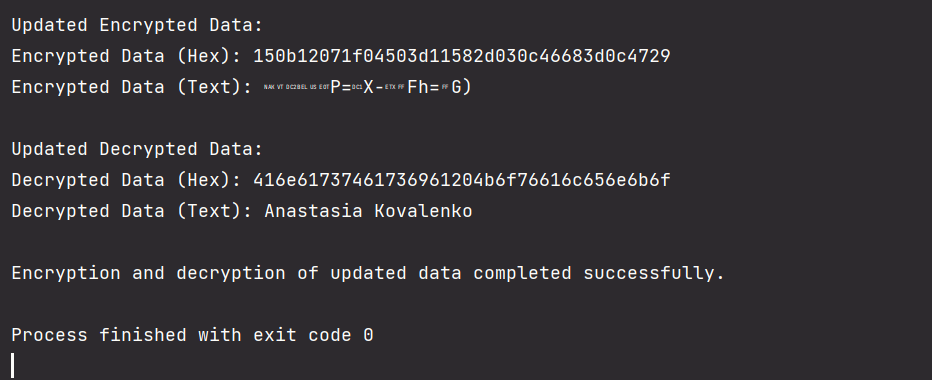


Рисунок 1 – Результат выполнения программы

**Вывод:** при выполнении работы были получены практические навыки решения задач целочисленного программирования.