Устимова Снежана, ПИ19-3

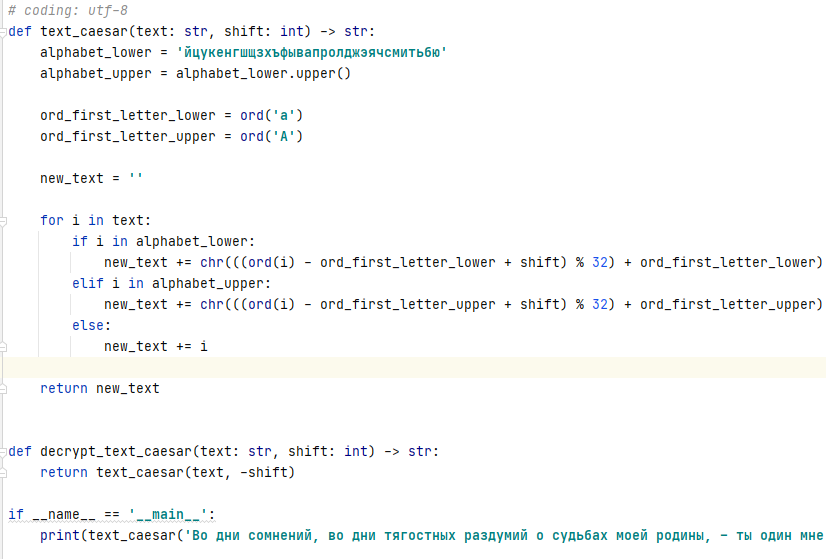
Лабораторная работа №6

Основные алгоритмы синхронного шифрования

Задания для выполнения

1. Написать функцию шифрования и дешифрования текста обобщенным шифром Цезаря.
2. Написать функцию, принимающую шифротекст, зашифрованный шифром из предыдущего задания и восстанавливающий текст, без знания ключа.
3. Реализовать в виде функций шифр Вернама.

1. Код программы:



(там для шифрования выбрано стихотворение в прозе про русский язык, потом покажу, почему я выбрала такую длинную строку).

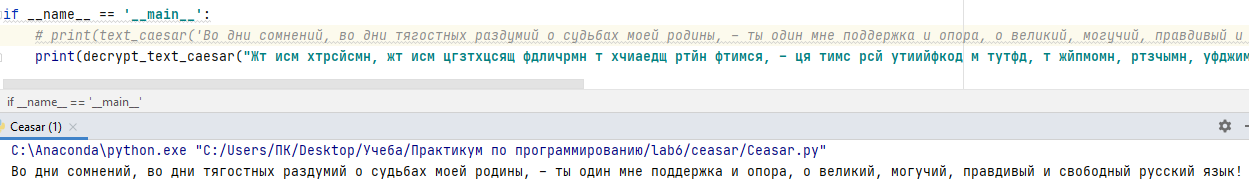
Запустим программу:



Вывело вот такой текст:

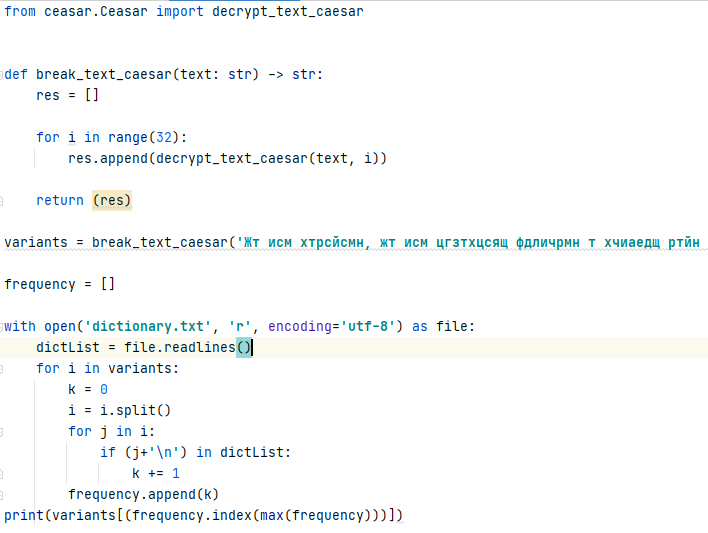
*Жт исм хтрсйсмн, жт исм цгзтхцсящ фдличрмн т хчиаедщ ртйн фтимся, – ця тимс рсй утиийфкод м тутфд, т жйпмомн, ртзчымн, уфджимжян м хжтетисян фчххомн гляо! Сй ечиа цйег – одо сй жудхца ж тцыдгсмй уфм жмий жхйзт, ыцт хтжйфьдйцхг итрд? Ст сйпалг жйфмца, ыцтея цдотн гляо сй еяп идс жйпмотрч сдфтич!*

Теперь попробуем дешифровать этот код:



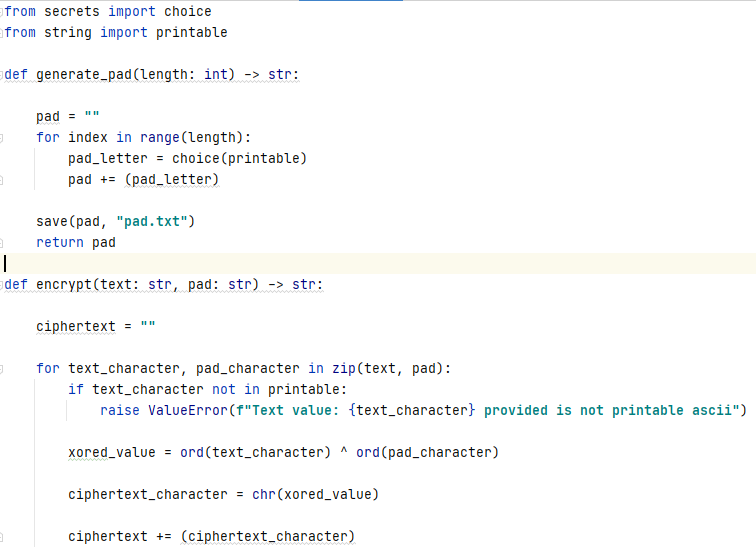
Все сработало. Едем дальше.

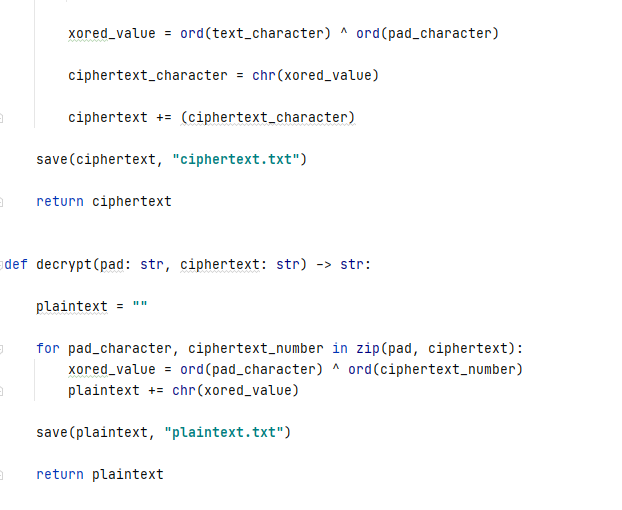
2. Код программы.

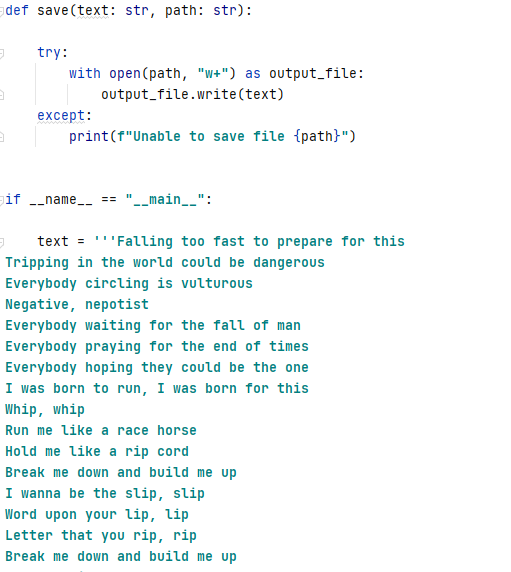


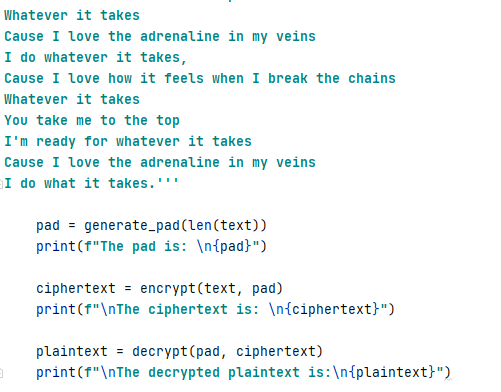
Объяснение: мы прогоняем нашу зашифрованную строку с каждым возможным смещением по алфавиту. То есть мы получаем список из всех возможных вариантов расшифрования строки при том или ином смещении. Затем я нашла в интернете 5000 самых частоупотребляемых слов в интернете, скопировала их в файл, после чего добавила этот файл на чтение в свое проект. Алгоритм: если в строке варианта есть слово, которое входит в этот список, то мы в список частот frequency добавляем 1, тем самым давая понять, что вот в этом варианте есть человеческое слово. Затем мы смотрим на этот список частот и замечаем, что в каком-то из них количество таких частот значительно выше остальных. Следовательно, это и есть наша изначальная строка. Взять ее можно по индексу максимальной частоты в списке.

3. Шифр Вернана = one time pad. Код программы:









**Здесь переменные:**

*Pad*: одноразовый pad, который используется для шифрования текста.

*Ciphertext*: текст, сгенерированный путем итеративного закрепления текста и пэда.

*Plaintext*: расшифрованное значение, сгенерированное с помощью XORing pad и шифротекста. Если все правильно, он должен соответствовать первоначальному тексту.

**Функции***:*

*Generate pad*: используется для генерации одноразовой панели. Будет принимать целое число для длины pad, которую он должен сгенерировать, и возвращать любой тип, наиболее подходящий для данного языка (либо строку, либо массив char).

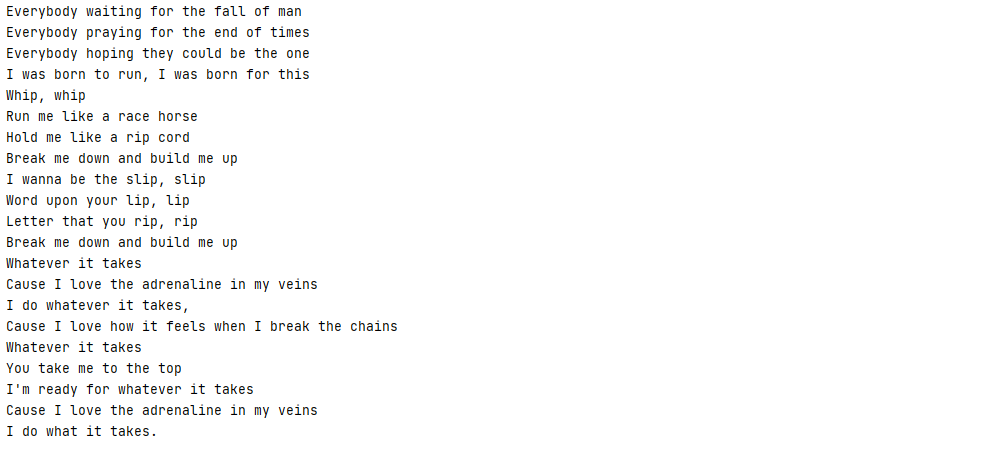
*Encrypt*: генерирует зашифрованный текст с помощью пэда и текста, а затем возвращает его как любой тип, наиболее подходящий для данного языка (либо строка, либо массив символов).

*Decrypt*: принимает зашифрованный текст и pad, генерирует открытый текст и возвращает его как любой тип, наиболее подходящий для языка (либо строка, либо массив символов).

*Save*: принимает либо строку, либо массив символов в зависимости от того, на каком языке он находится, а затем путь к файлу сериализует массив строк/символов в указанный путь.

Результат:



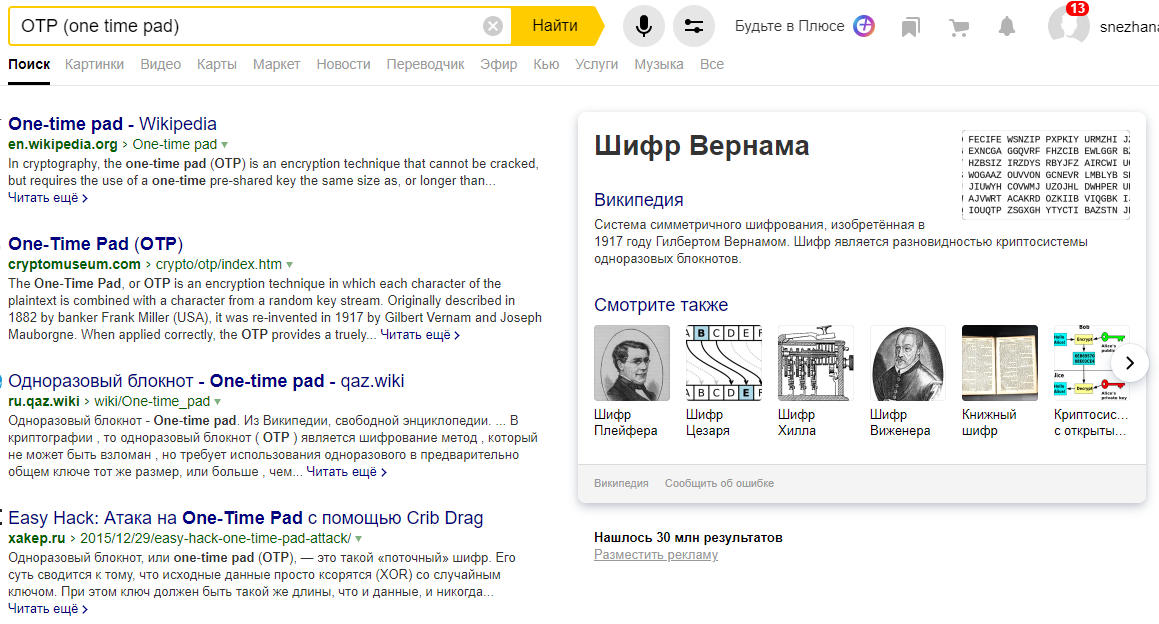


Все правильно, так как мы получили исходный текст.

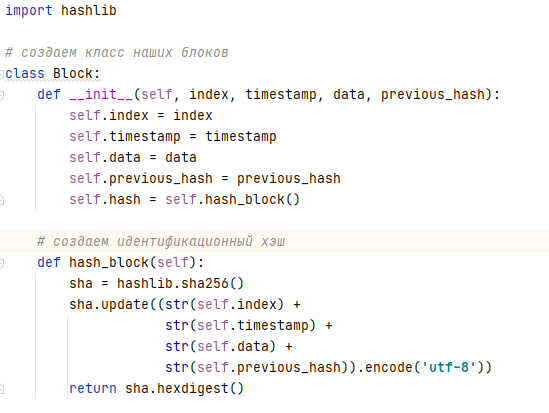
**Дополнительные задания**

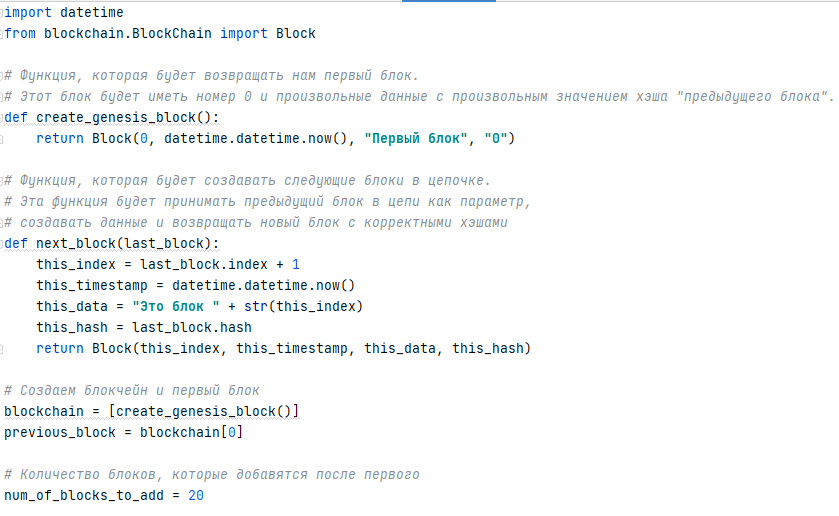
1. Реализуйте алгоритм шифрования OTP (one time pad).
2. Реализуйте алгоритм цепочки блоков (Cipher block chaining)

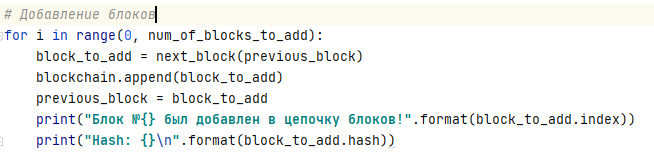
1. Яндекс говорит, что one time pad – это и есть шифр Вернама, а он уэе реализован.



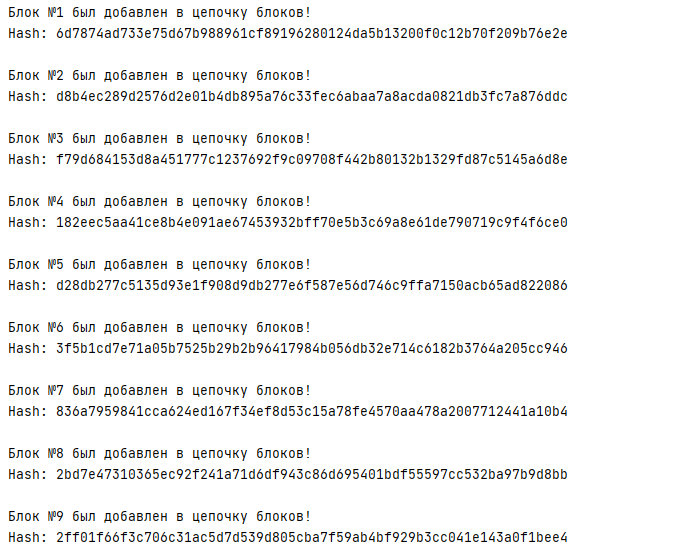
2. Все комментарии на этот раз писала прямо в коде.







Запустим:



И т.д…