# Лабораторная работа 2

## Тема:

“Защитное кодирование по методу Хэмминга”

## Выполнил:

студент группы 22207 Гордеев Никита

## Задание:

* Составить программу, которая будет выполнять кодирование методом Хэмминга произвольной битовой последовательности заданной вариантом длины (задается в виде последовательности символов 0 и 1, вводимых без пробелов).

## Значения параметров разбиения (Вариант 1):

* Длина кодируемой последовательности: 9
* Длина новой последовательности: 13

## О методе:

* Общепринятой практикой в области облачных технологий является хранение информации (данных) у одного поставщика услуг в одном физическом и географическом местоположении. При этом можно задействовать внушительный набор инструментов по организационным и техническим мерам обеспечения защиты информации в «облаке», что, как правило, сводит на нет преимущества облачного хранилища. Поэтому ценная критически важная информация обязательно шифруется перед её отправкой в «облако» на стороне пользователя. Для этого можно использовать средства как простые (например, создание архивов, защищенных паролем), так и более продвинутые (ПО для создания шифрованных разделов). Для пользователя помимо обычного кодирования своей информации, существуют методы разделения информации, которые позволяют разделить их сообщение и отправить его по разным внешним хранилищам, что существенно лучше, чем просто отправить оригинал с резервными копиями по разным хранилищам. К таким методам можно отнести, например, метод рассечения-разнесения
* Рассмотренный же в данной статье метод рассечения разнесения позволяет разбить исходный файл на несколько частей и каждую хранить в разных местах хранения и если злоумышленник сможет добыть одну часть и успешно взломать её, то достичь цели крипто атаки всё равно не сможет.
* Набор защищаемых данных разбивается на блоки, которые разносятся по нескольким другим наборам данных.
* Каждый отдельный блок не несет сколько-нибудь значимой информации, и даже доступ к полной совокупности блоков не позволяет легко восстановить исходный набор данных без знания способа разбиения.

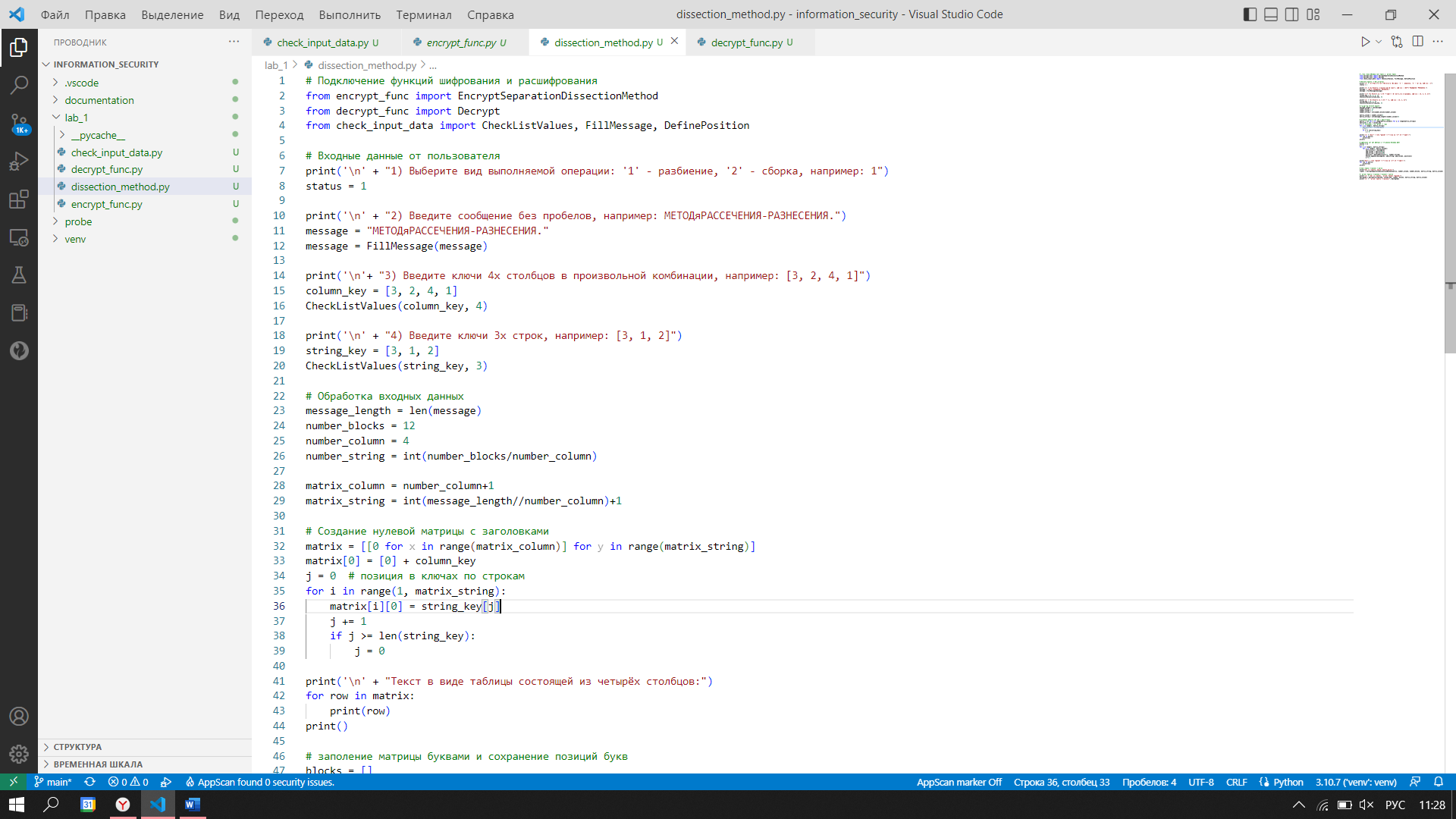
## Решение:

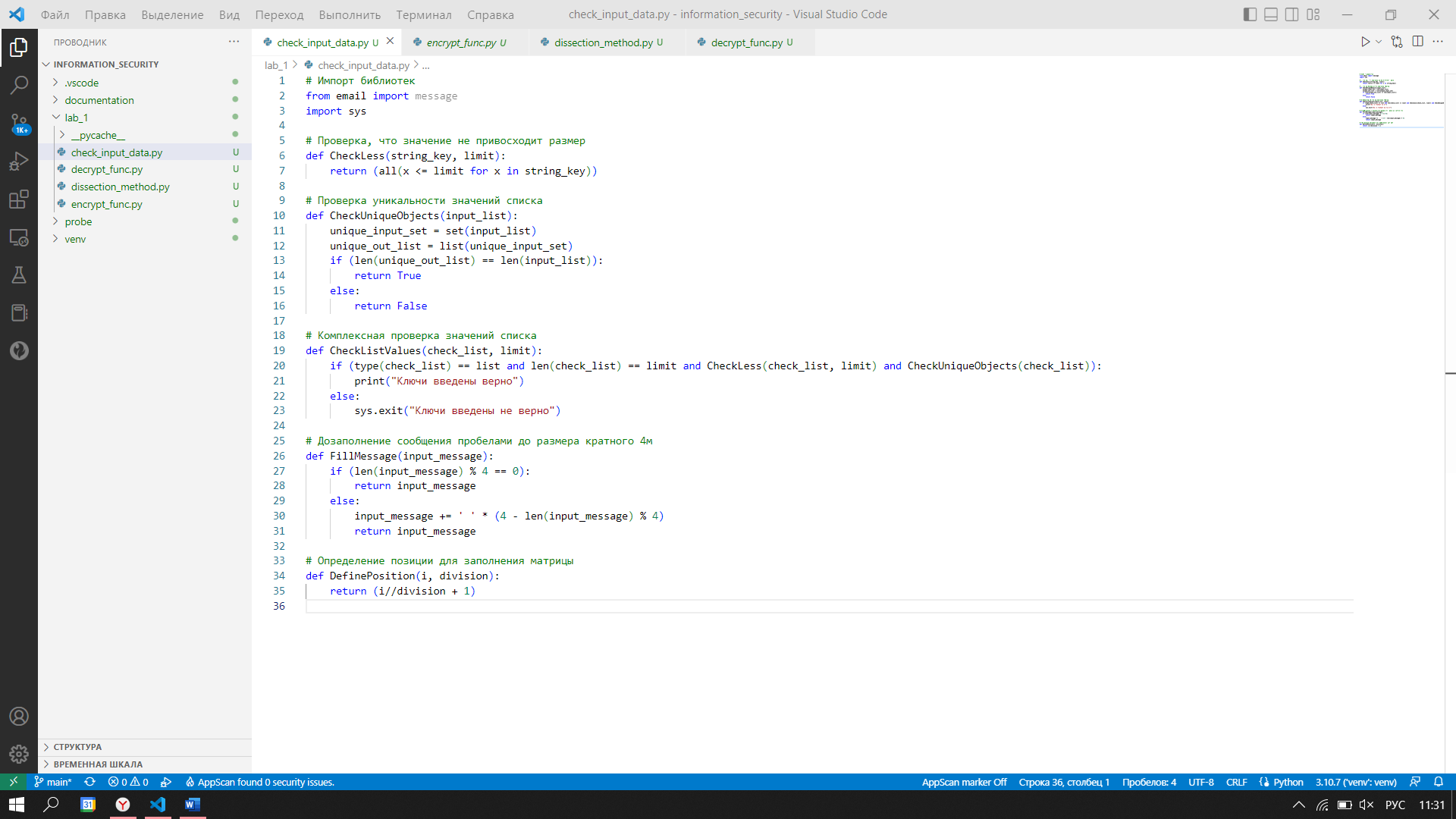
1) Длина N новой последовательности должна быть не менее 13, так как резервные 4 позиции позволяют получить двоичное представление чисел от 0 до 15, что достаточно для отображения любого возможного места ошибки.

1) Решение на бумаге

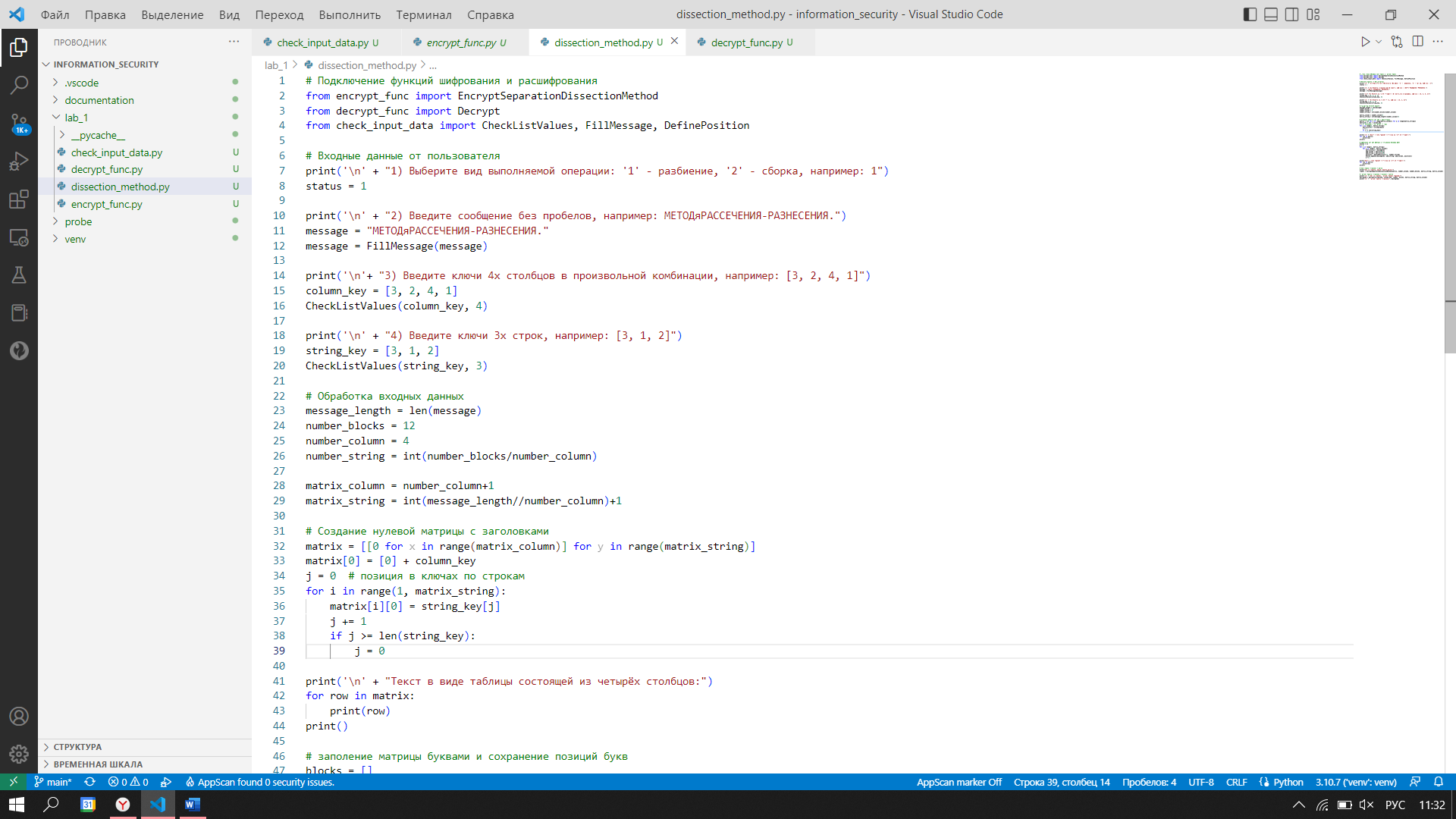
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ключи | 3 | 2 | 4 | 1 |  | n |  |
| 3 | М | Е | Т | О |  | 4 |  |
| 1 | Д | я | Р | А |  |  |  |
| 2 | С | С | Е | Ч |  | Номер блока | Содержимое |
| 3 | Е | Н | И | Я |  | 1 | АЗ |
| 1 | - | Р | А | З |  | 2 | яР |
| 2 | Н | Е | С | Е |  | 3 | Д- |
| 3 | Н | И | Я | . |  | 4 | РА |
|  |  |  |  |  |  | 5 | ЧЕ |
| K | 11 | 10 | 12 | 9 |  | 6 | СЕ |
| 3 | 2 | 4 | 1 |  | 7 | СН |
| 7 | 6 | 8 | 5 |  | 8 | ЕС |
| 11 | 10 | 12 | 9 |  | 9 | ОЯ. |
| 3 | 2 | 4 | 1 |  | 10 | ЕНИ |
| 7 | 6 | 8 | 5 |  | 11 | МЕН |
| 11 | 10 | 12 | 9 |  | 12 | ТИЯ |

## 2) Алгоритм программы:

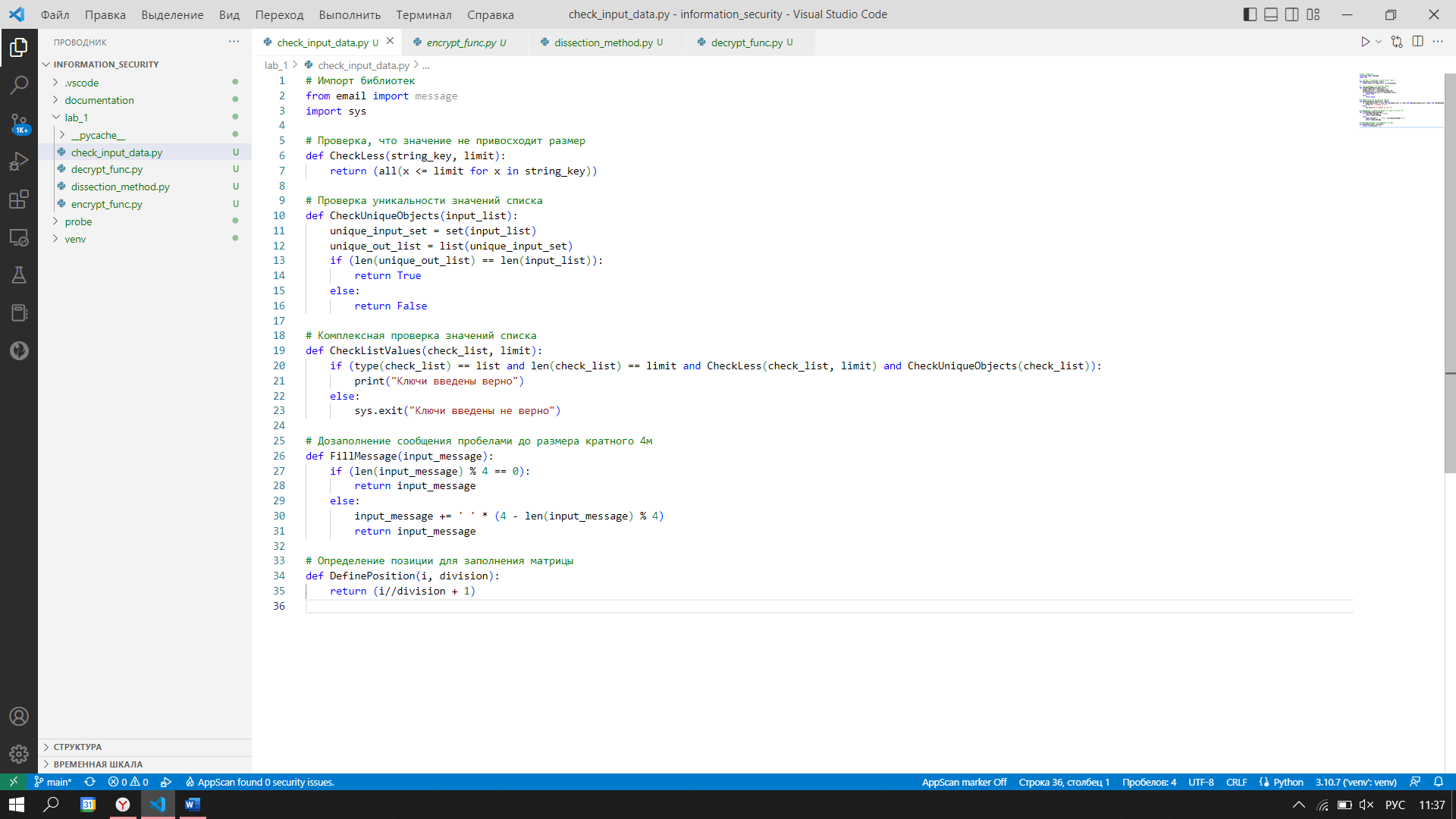
1. Получение сведений о программе
   1. Получение сообщения
      1. Проверка на корректность введённых данных



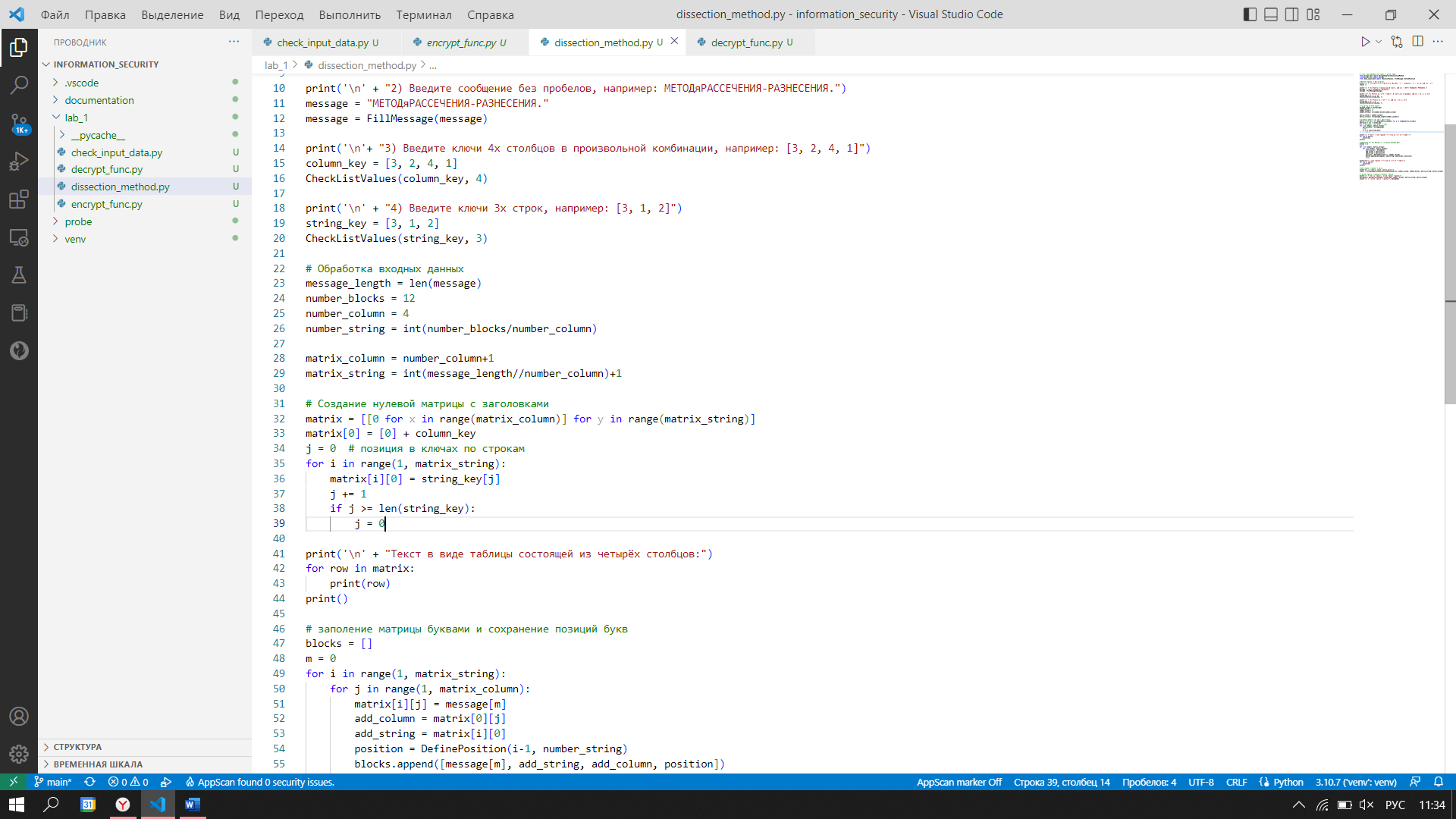
* 1. Получение ключей столбцов и строк



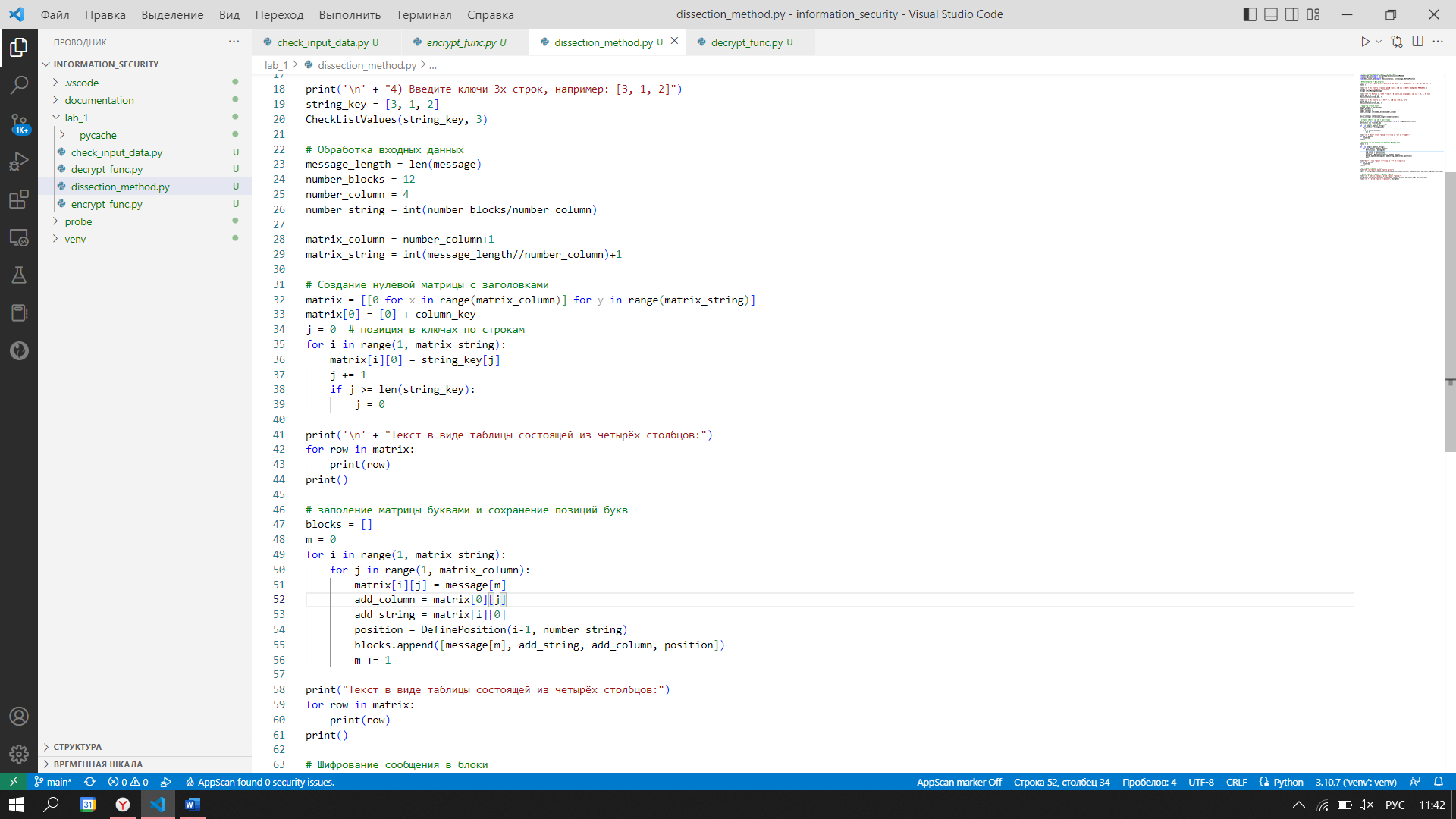
* + 1. Проверка на корректность введённых данных

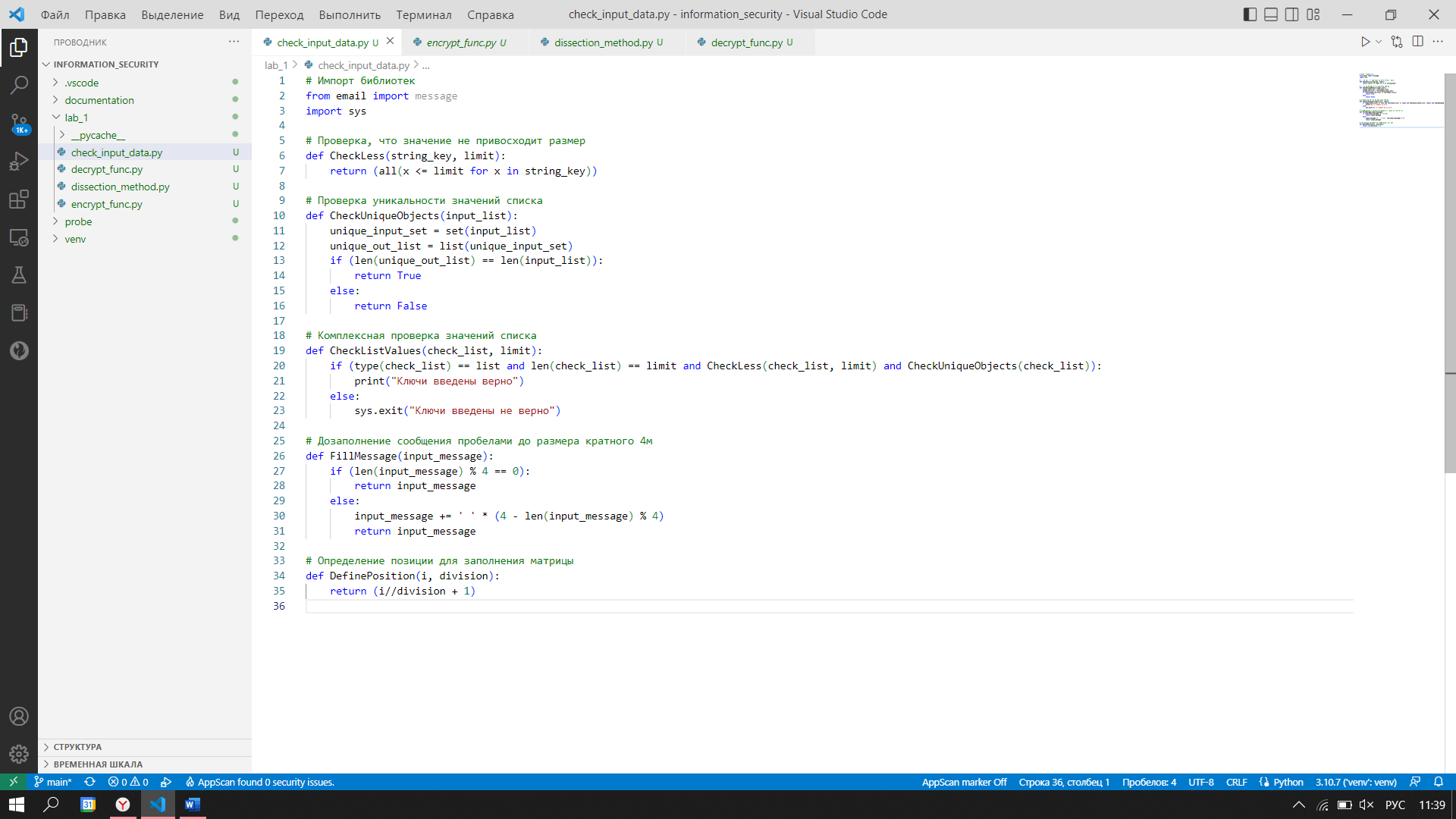


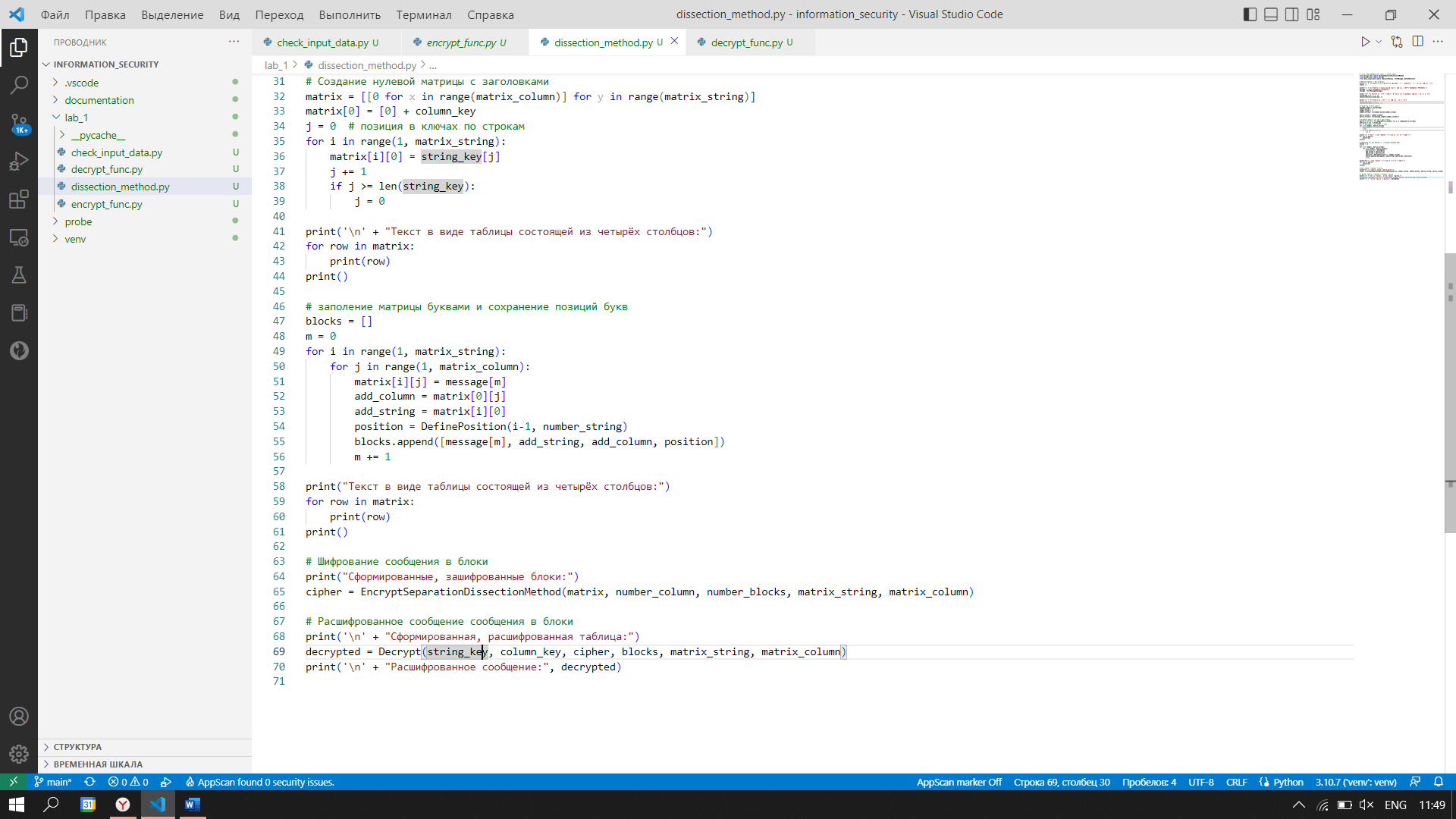
* 1. Получение количества строк и столбцов

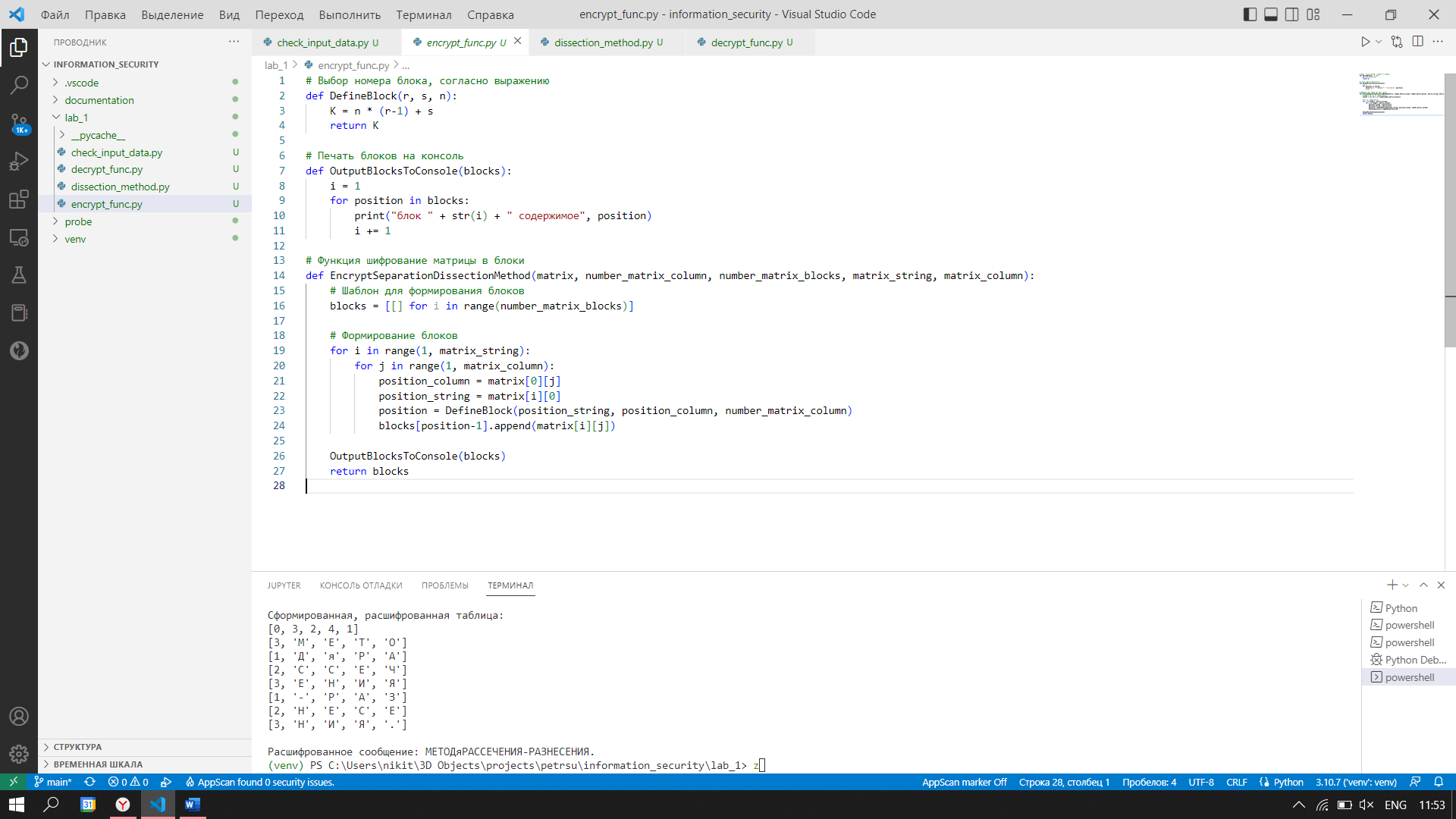


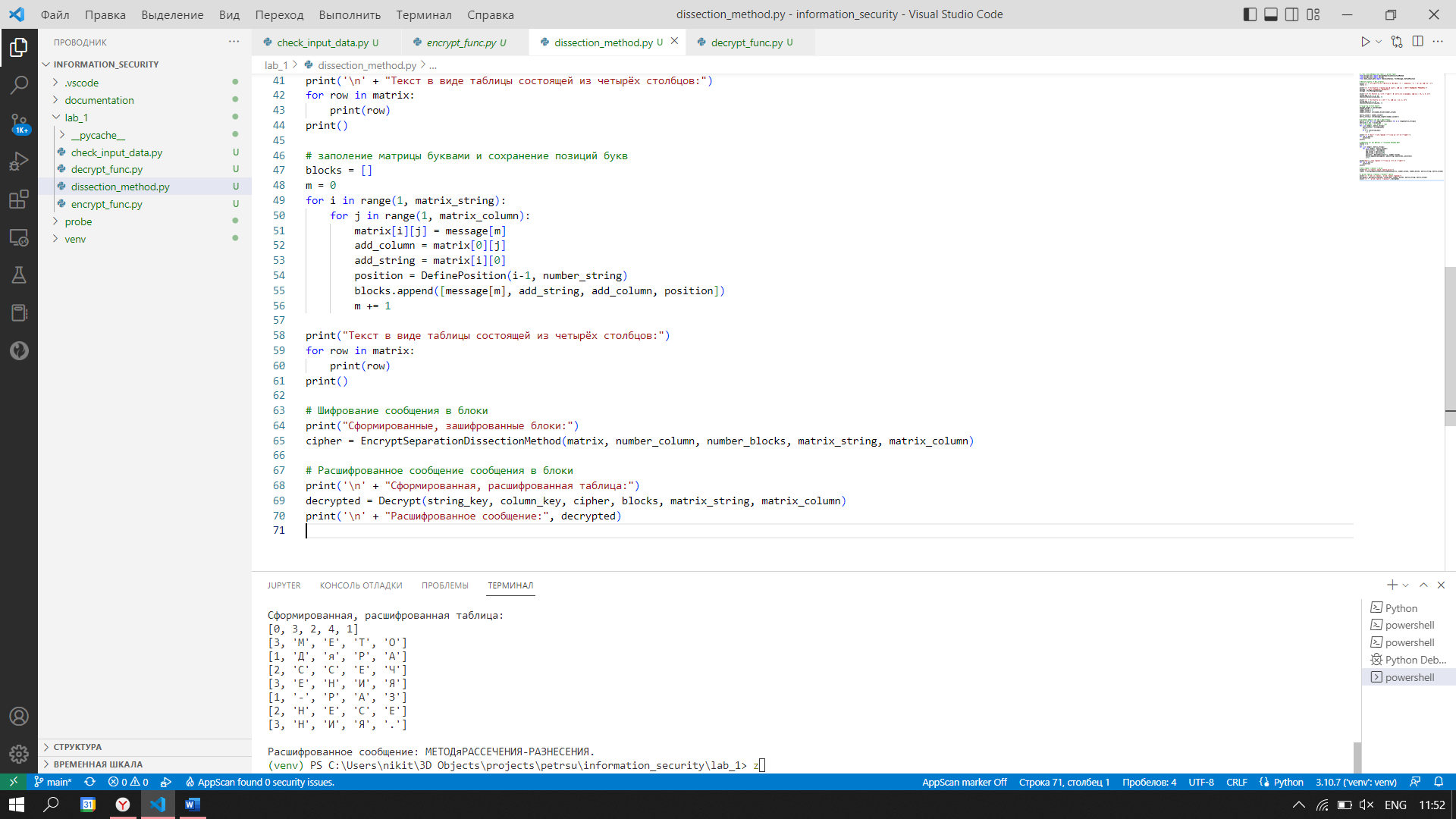
1. Шифрование сообщения
   1. Создание нулевой матрицы с заголовками



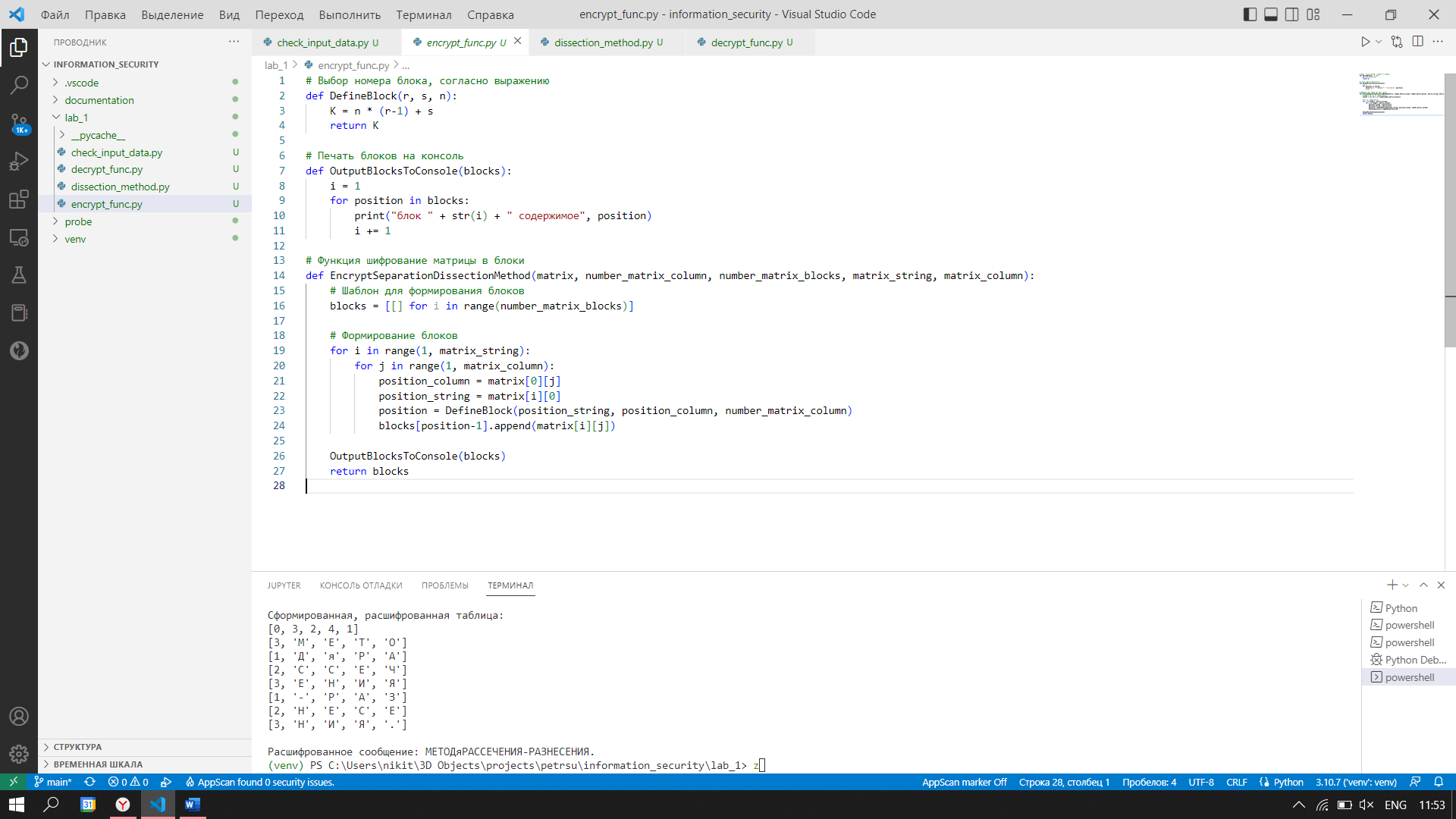
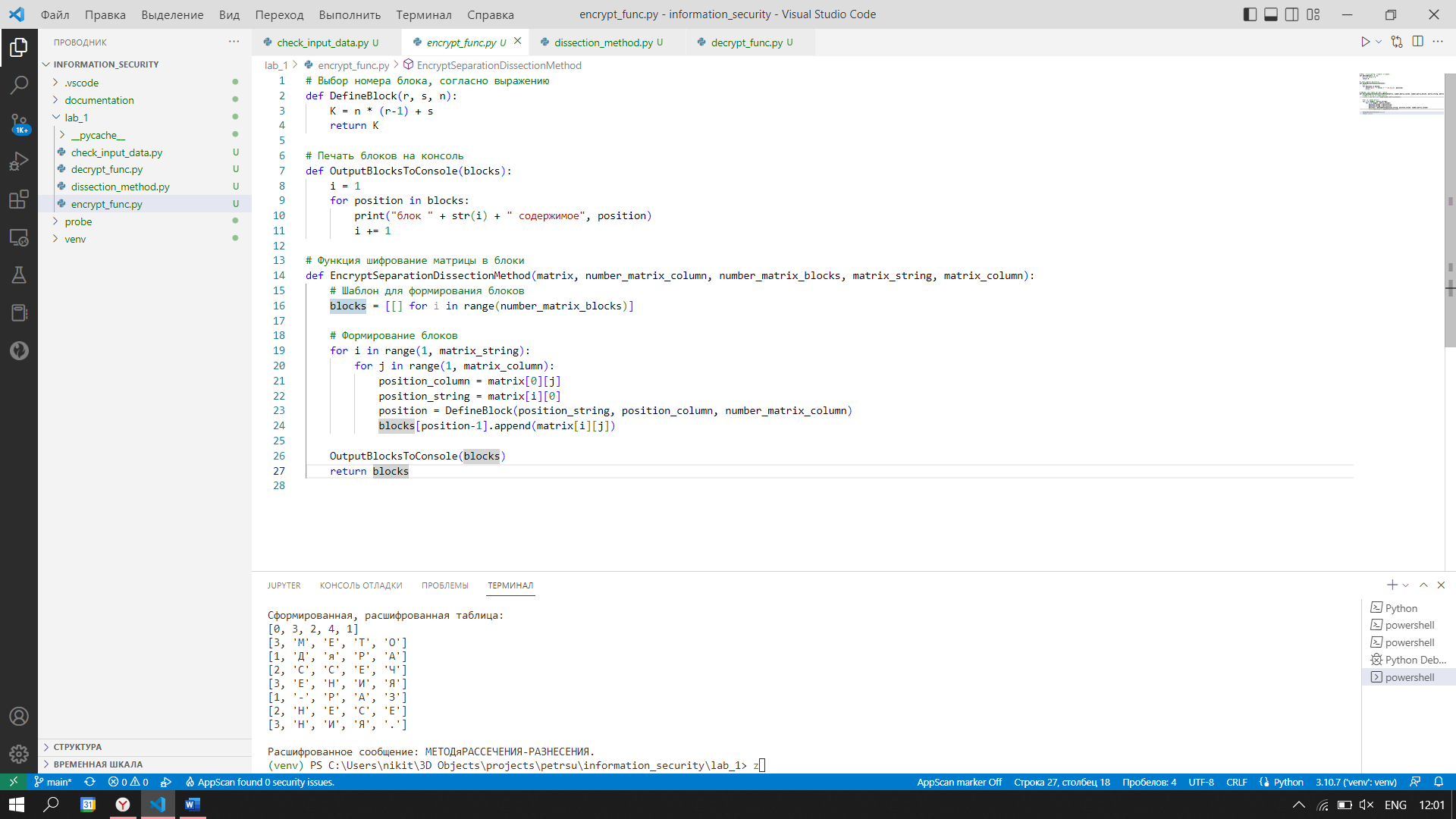
* 1. Заполнение матрицы буквами и сохранение позиций букв



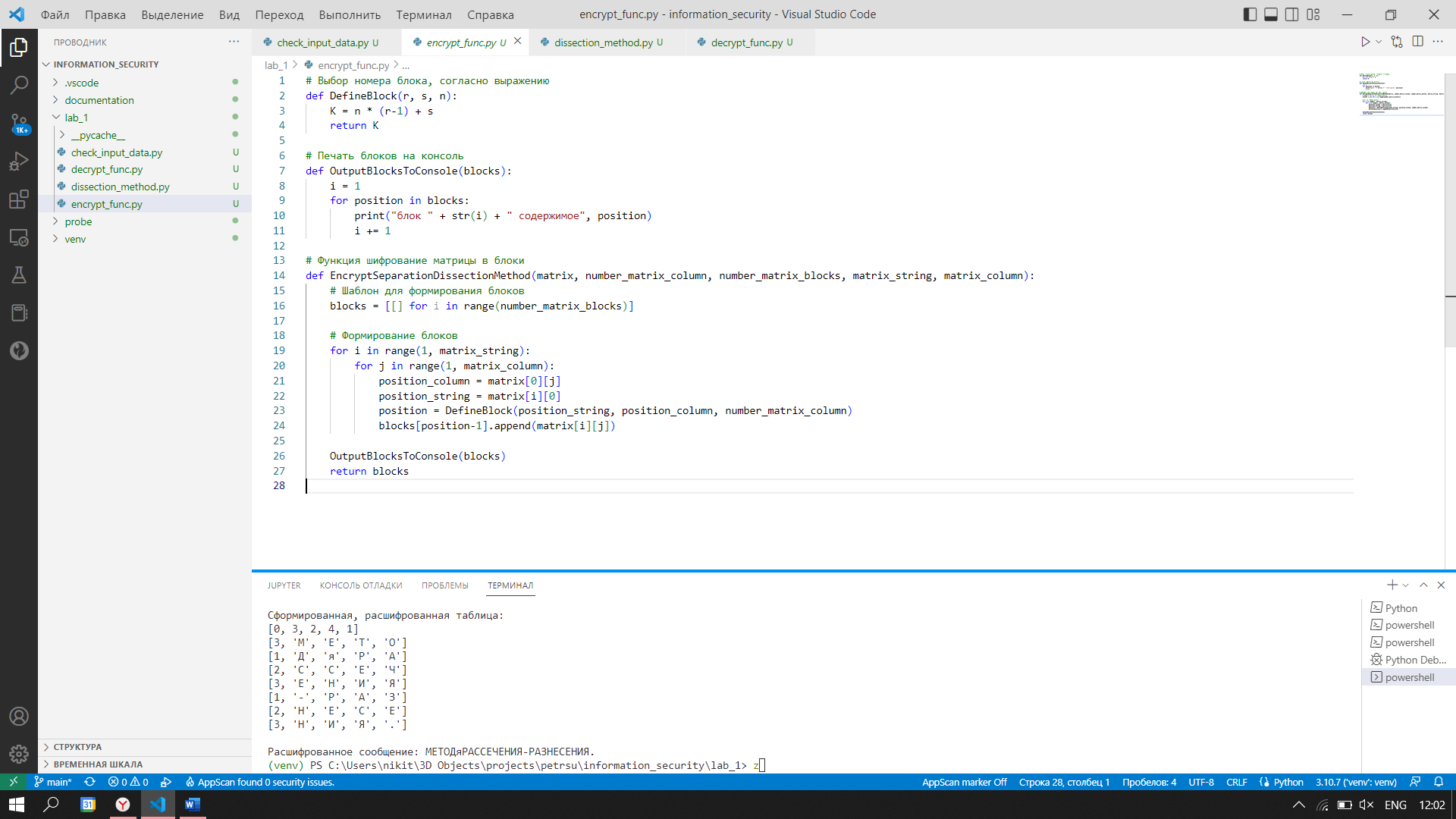
* 1. Создание матрицы блоков

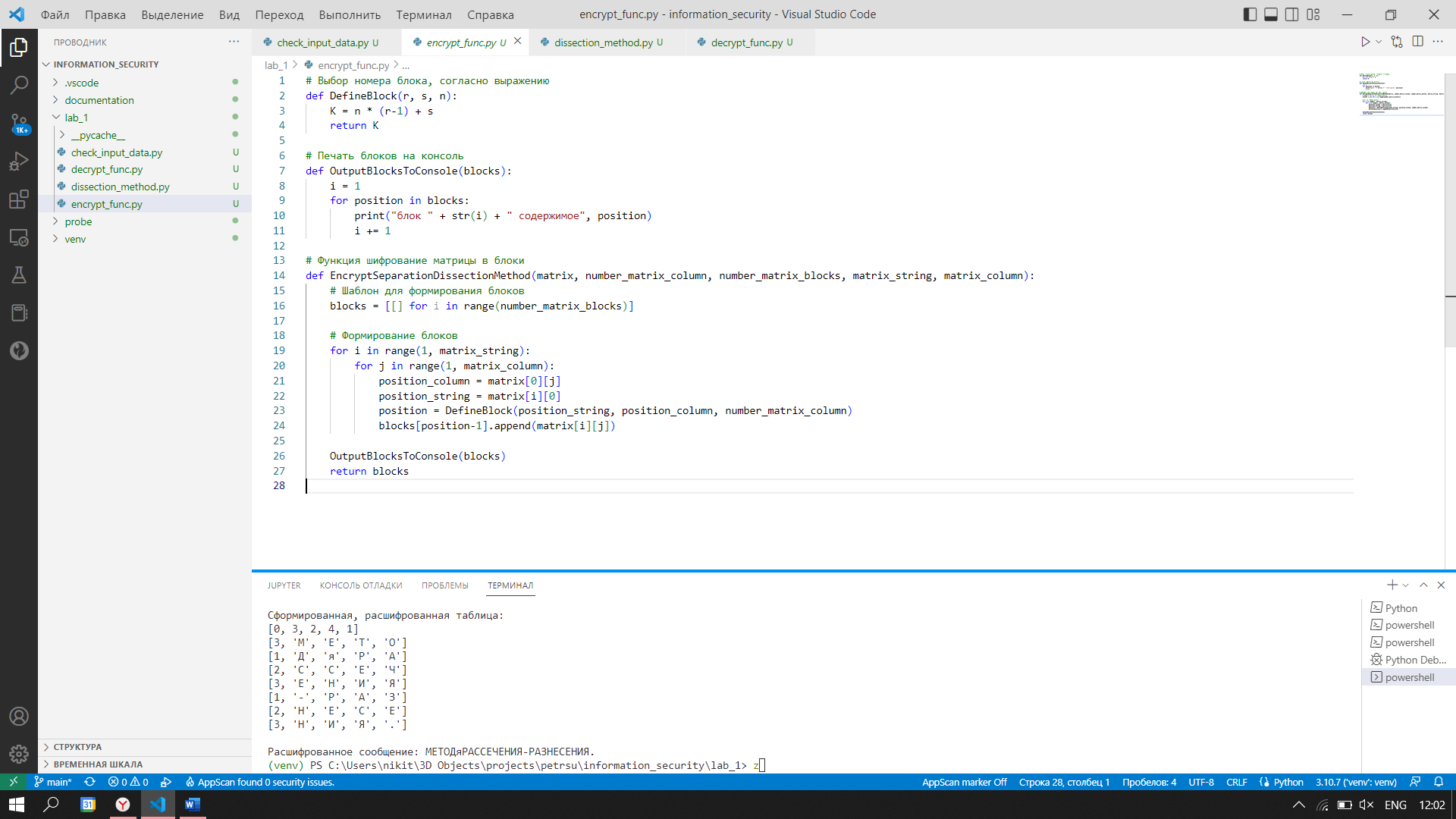


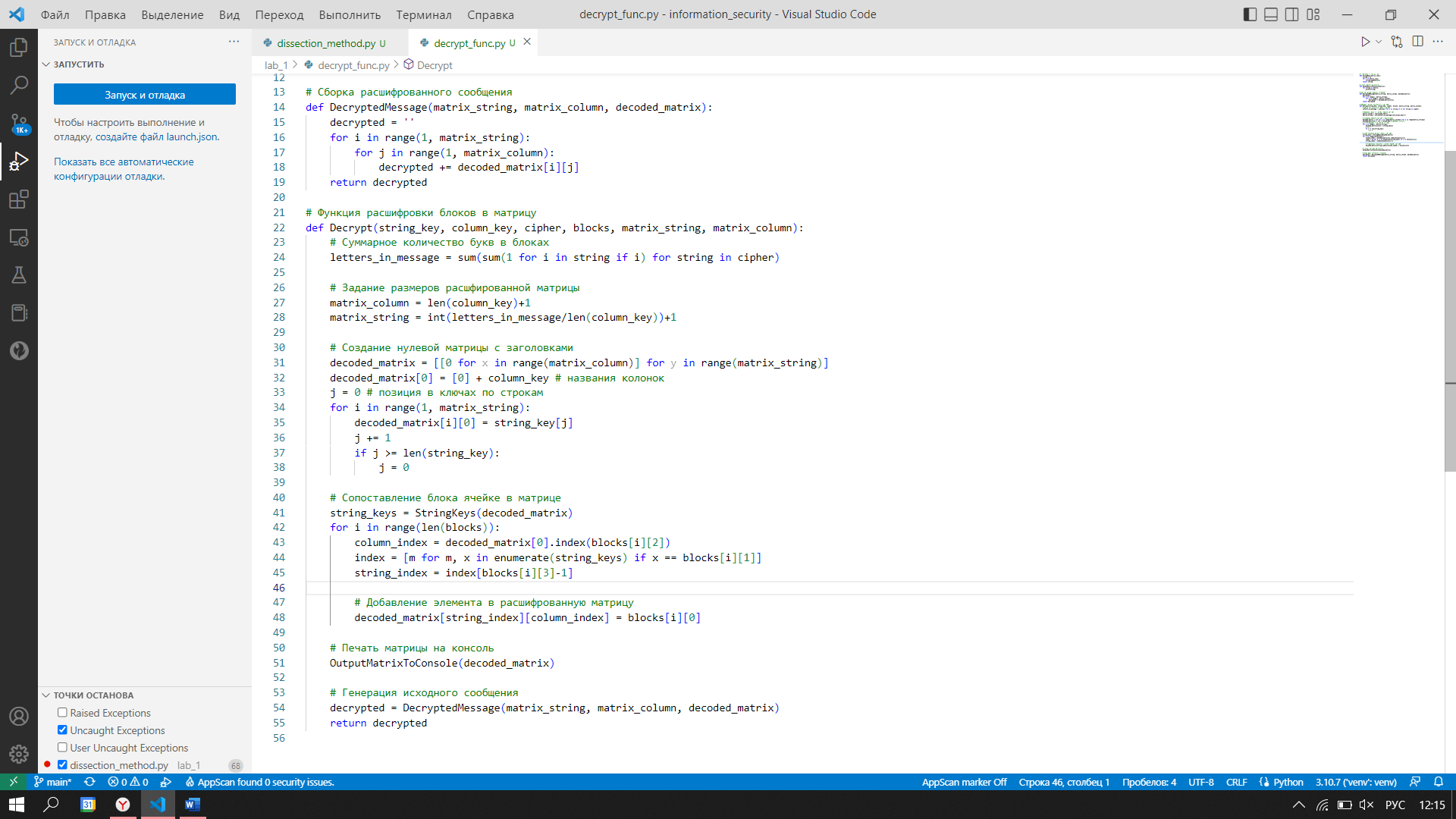
* 1. Распределение букв из матрицы в блоки

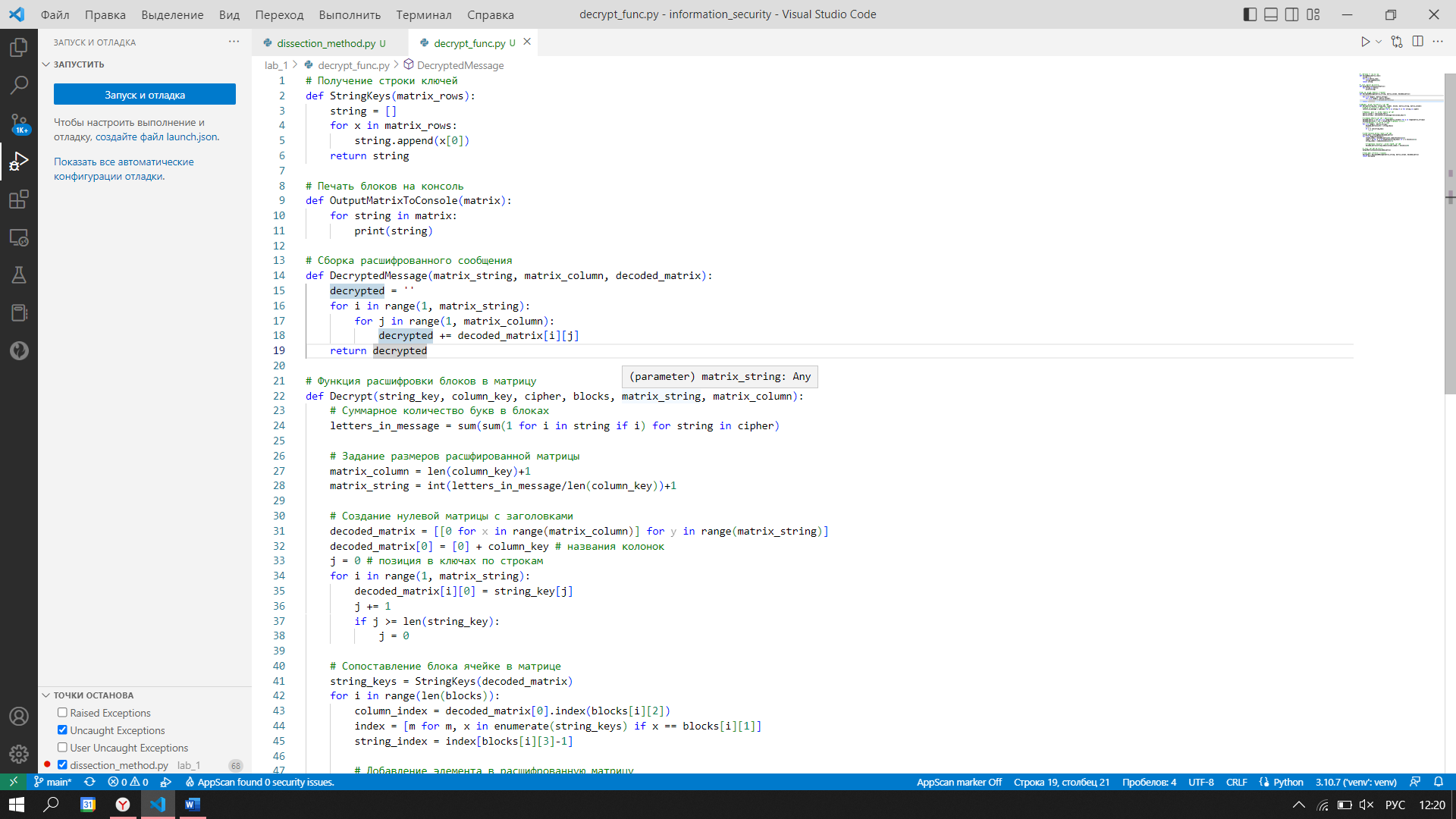


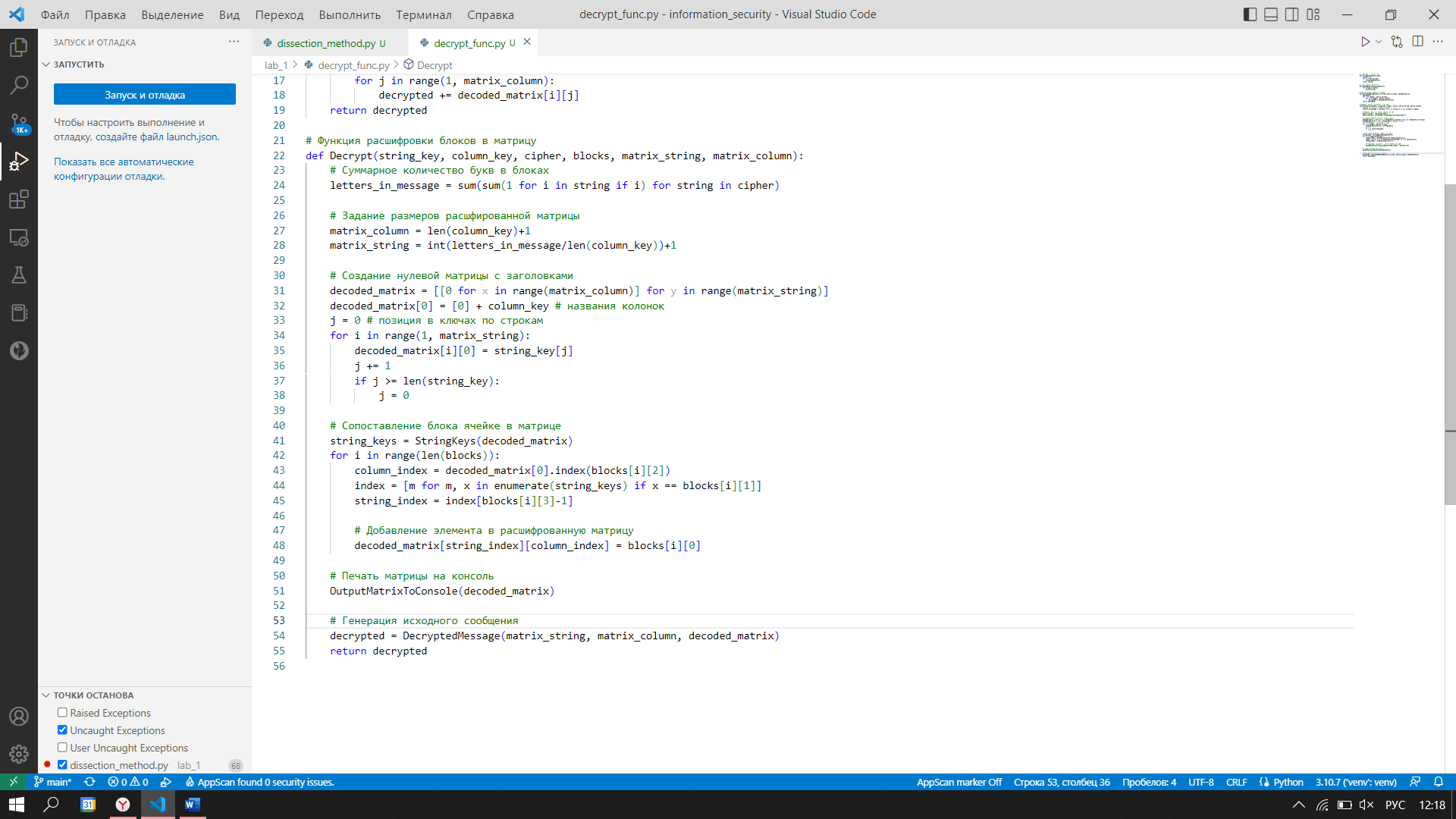
* 1. Возврат блоков



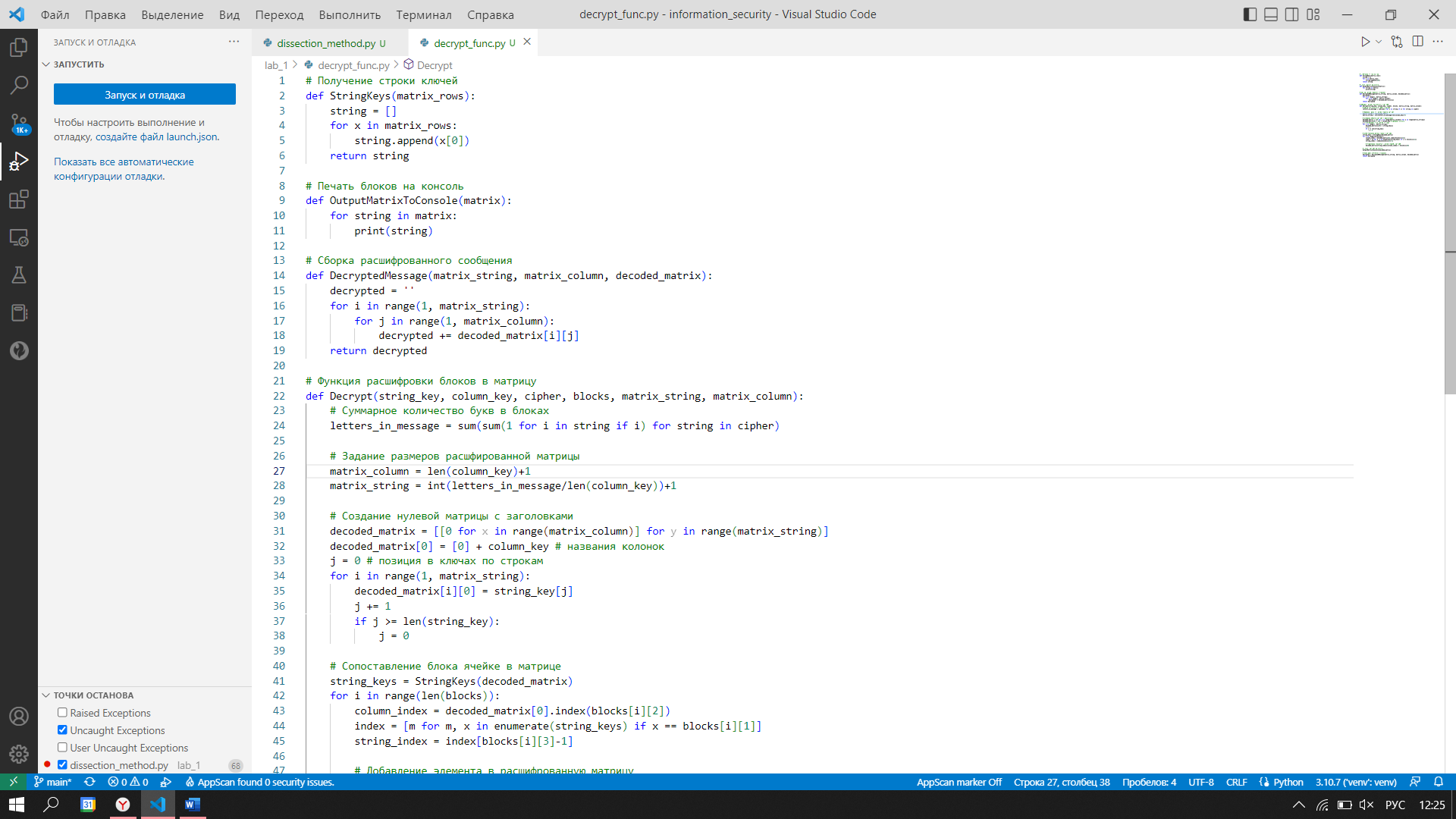


1. Расшифрование блоков
   1. Создание матрицы с заголовками
   2. Распределение букв в блоках по ячейкам матрицы

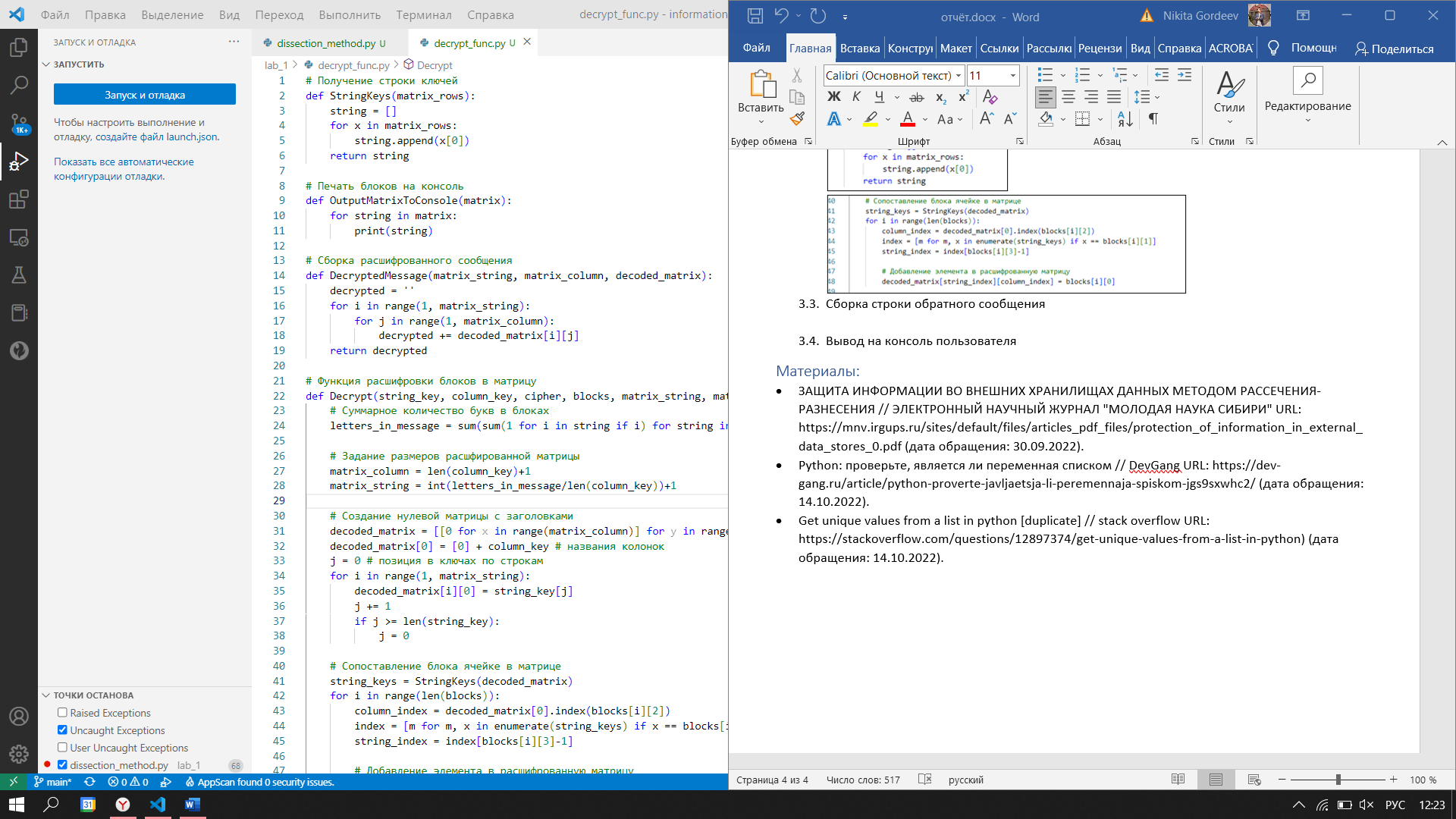
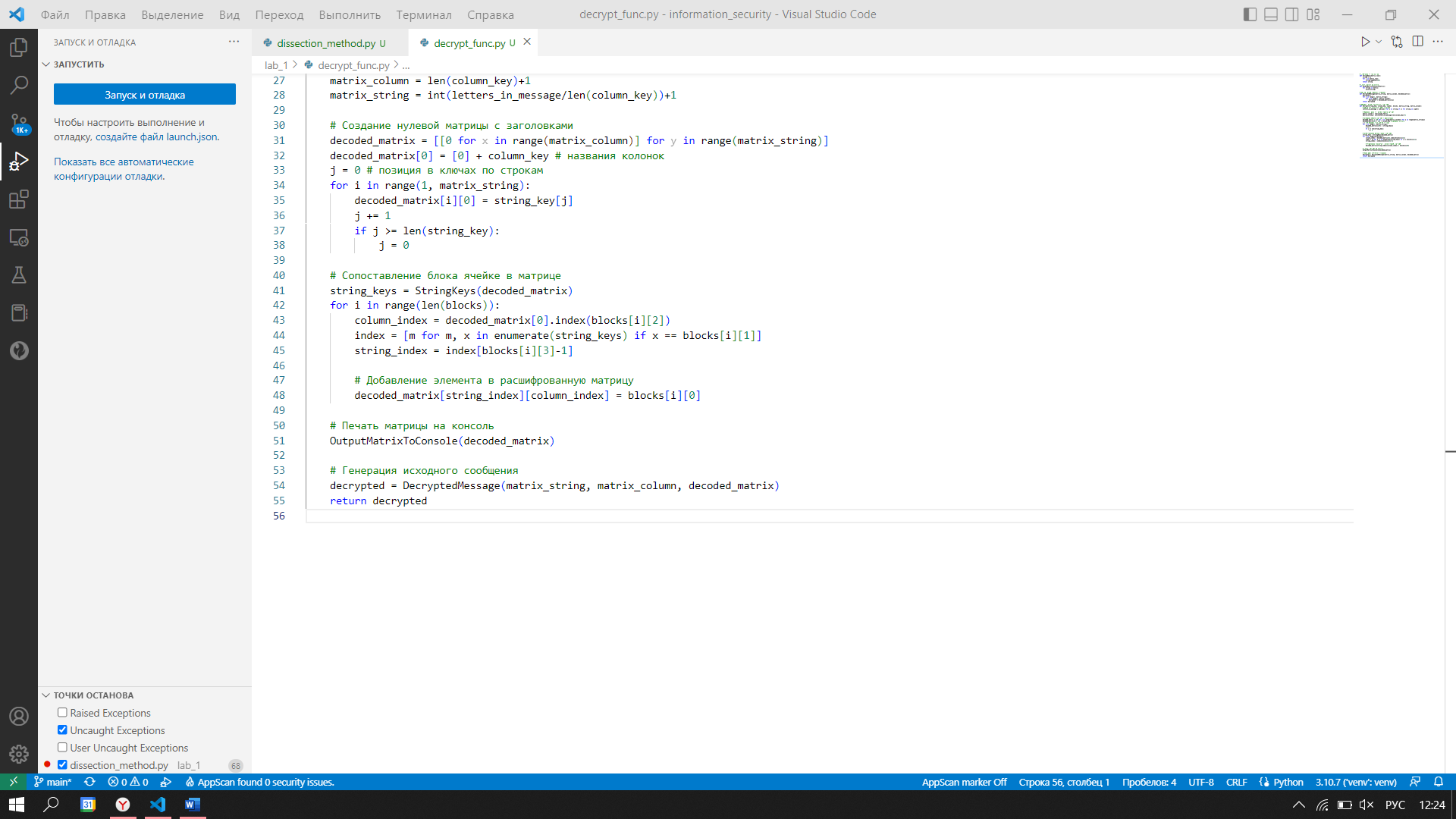




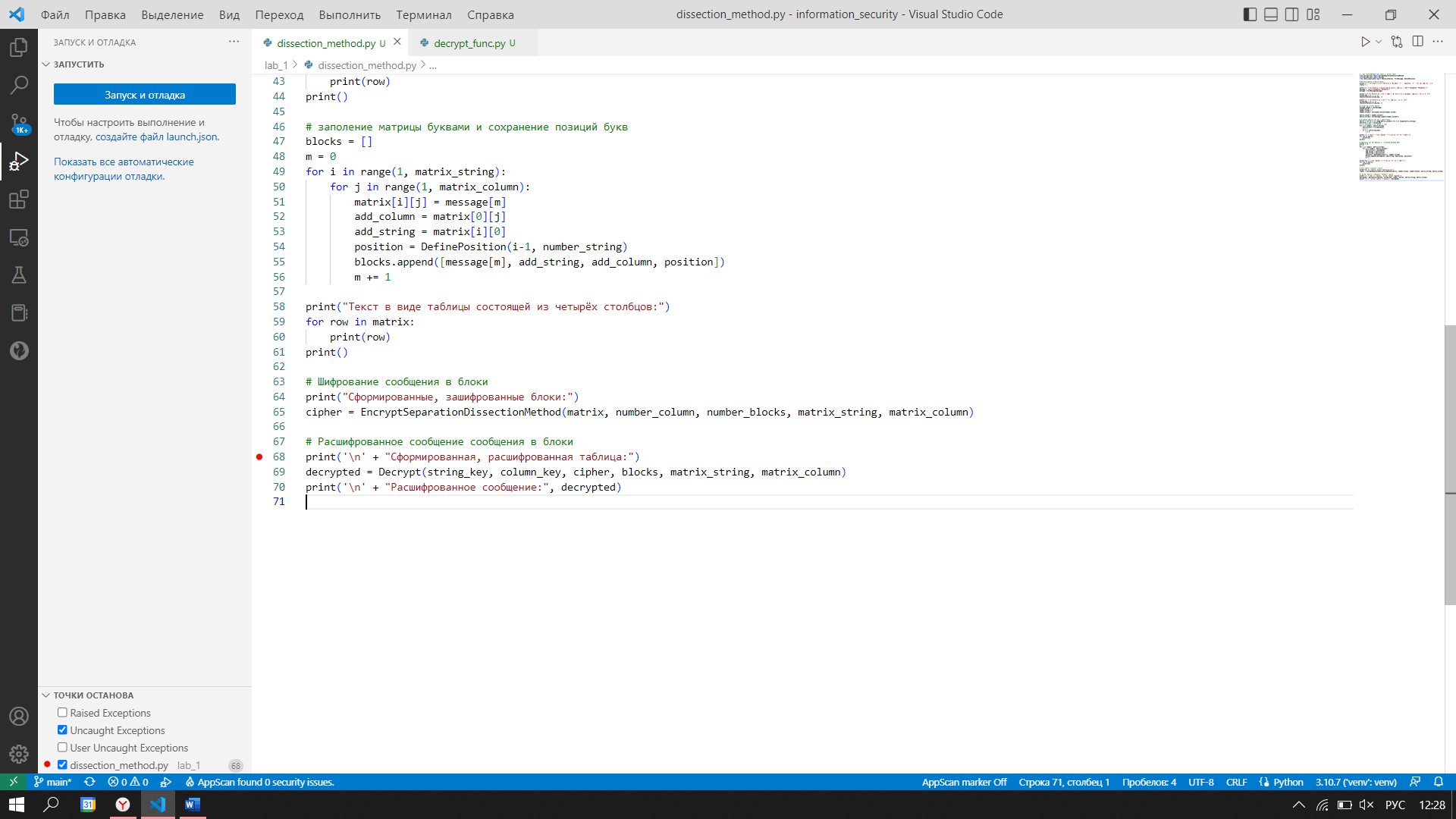
* 1. Печать расшифрованной матрицы на консоль



* 1. Сборка строки обратного сообщения



* 1. Вывод на консоль пользователя



## Материалы:

* ЗАЩИТА ИНФОРМАЦИИ ВО ВНЕШНИХ ХРАНИЛИЩАХ ДАННЫХ МЕТОДОМ РАССЕЧЕНИЯ-РАЗНЕСЕНИЯ // ЭЛЕКТРОННЫЙ НАУЧНЫЙ ЖУРНАЛ "МОЛОДАЯ НАУКА СИБИРИ" URL: https://mnv.irgups.ru/sites/default/files/articles\_pdf\_files/protection\_of\_information\_in\_external\_data\_stores\_0.pdf (дата обращения: 12.10.2022).
* Python: проверьте, является ли переменная списком // DevGang URL: https://dev-gang.ru/article/python-proverte-javljaetsja-li-peremennaja-spiskom-jgs9sxwhc2/ (дата обращения: 12.10.2022).
* Get unique values from a list in python [duplicate] // stack overflow URL: https://stackoverflow.com/questions/12897374/get-unique-values-from-a-list-in-python) (дата обращения: 13.10.2022).

'001100010

011000000

110001001

100010011

000000110

001001100

000011000

100110000'