Решение.

1. Расставим степени вершин графа

**4**

1. однозначно можно определить точки лежащие на оси симметрии, Б (единственная четверка которая имеет соседей 2,2,4,4), Д (единственная четверка которая имеет соседей 4,4,3,3), И (единственная двойка которая имеет соседей 3,3).



**2**

**4**

**2**

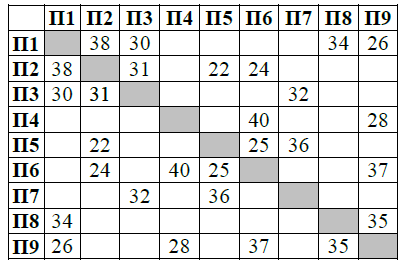
**4**

**3**

**2**

**3**

**4**



**4**

**4**

**3**

**2**

**3**

**4**

**2**

**2**

**4**

* 1. Б - П9
  2. Д - П2
  3. И – П7

**Ж**

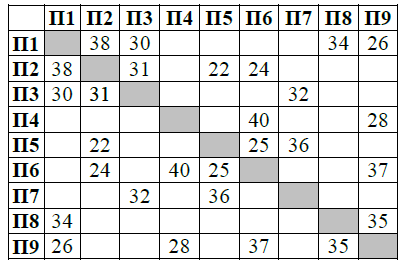
**А**

**Г**

**В**

**К**

**Е**



**4**

**4**

**3**

**2**

**3**

**4**

**2**

**2**

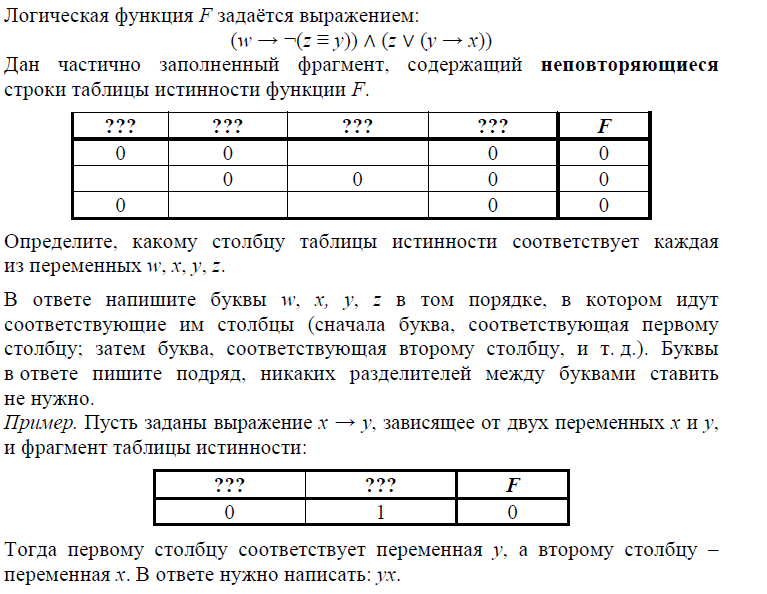
**4**

**Б**

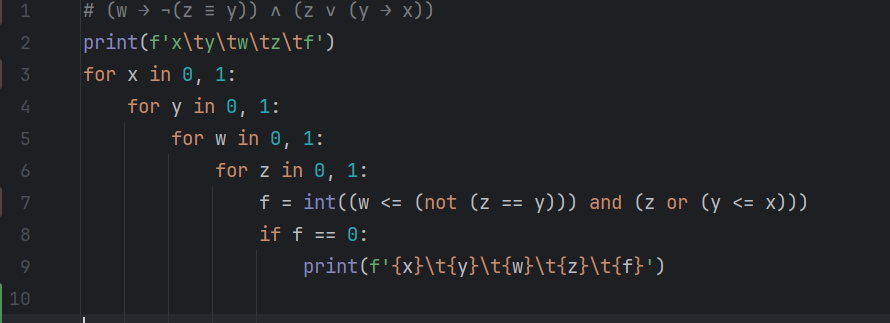
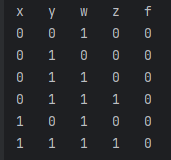
**И**

**Д**

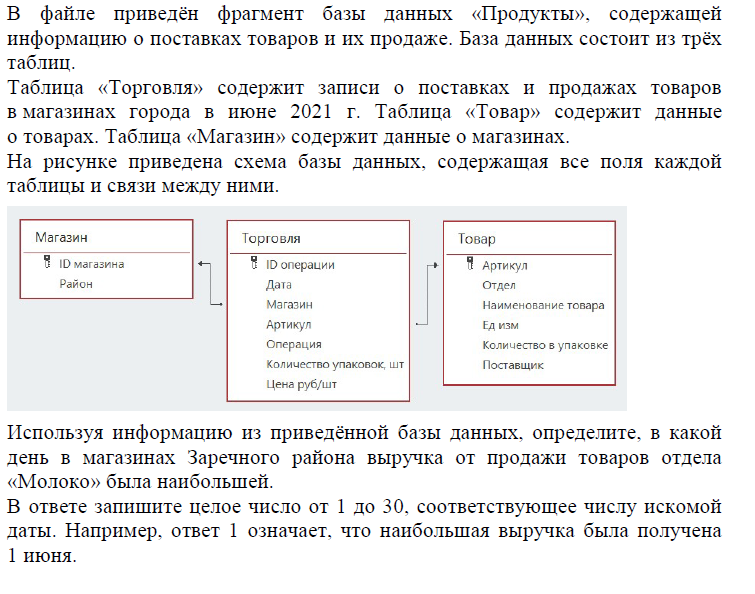
1. Предположим вершина Е это П1, тогда:
   1. В – П8
   2. К – П3
   3. Г – П6
   4. А –П4
   5. Ж – П3
2. Найдем длину дорог КИ и АГ:
   1. КИ = 32
   2. АГ = 40
3. Условие КИ <АГ выполнено, значит наше предположение верно.
4. ДЖ = 22



**Решение**.

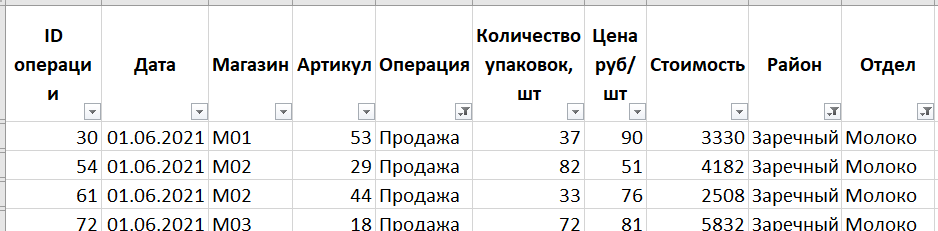
1. Напишем программу для составления части таблицы истинности:
2. Получаем результат:
3. В исходной таблице не может быть строки где все «1», удалим ее из ответа
4. W точно должна стоять в 3 столбце, только у этой переменной один «0»
5. Y точно должна стоять в 1 столбце, так как в исходной таблице вторая строка \_ 0 0 0.
6. Из нашей таблицы можно удалить 3 и 4 строку:

Ответ **YXWZ**

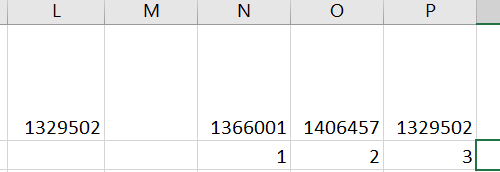


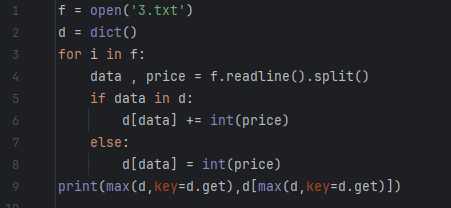
**Решение**

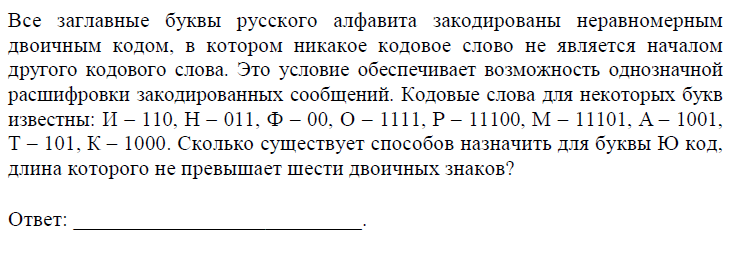
1. Добавим в таблицу Торговля столбцы:
   1. Стоимость: =F2\*G2
   2. Район: =ВПР(C2;Магазин!A:B;2;0)
   3. Отдел: =ВПР(D2;Товар!A:F;2;0)
2. Установим нужные фильтры:



1. Так как магазинов в районе много, одним из вариатор решения будет использовать формулу =ПРОМЕЖУТОЧНЫЕ.ИТОГИ(9;H:H) и просмотреть каждую дату, но это долго, лучший вариант скопировать в текстовый файл два столбца Дата и Стоимость, а затем посчитать программой

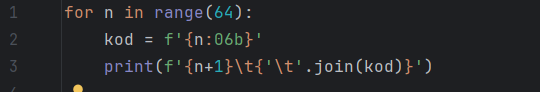




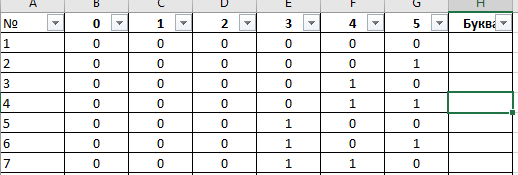


**Решение**

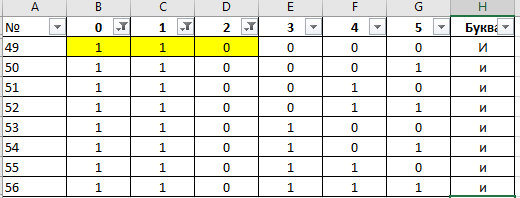
Построим всевозможные комбинации 6 битных кодовых цепочек. Для этого можно написать программу:



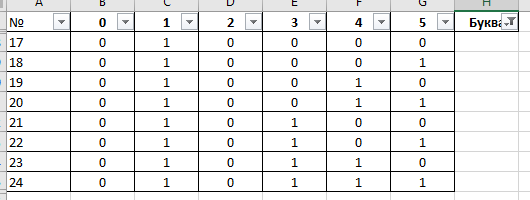
Перенесем результат в Excel



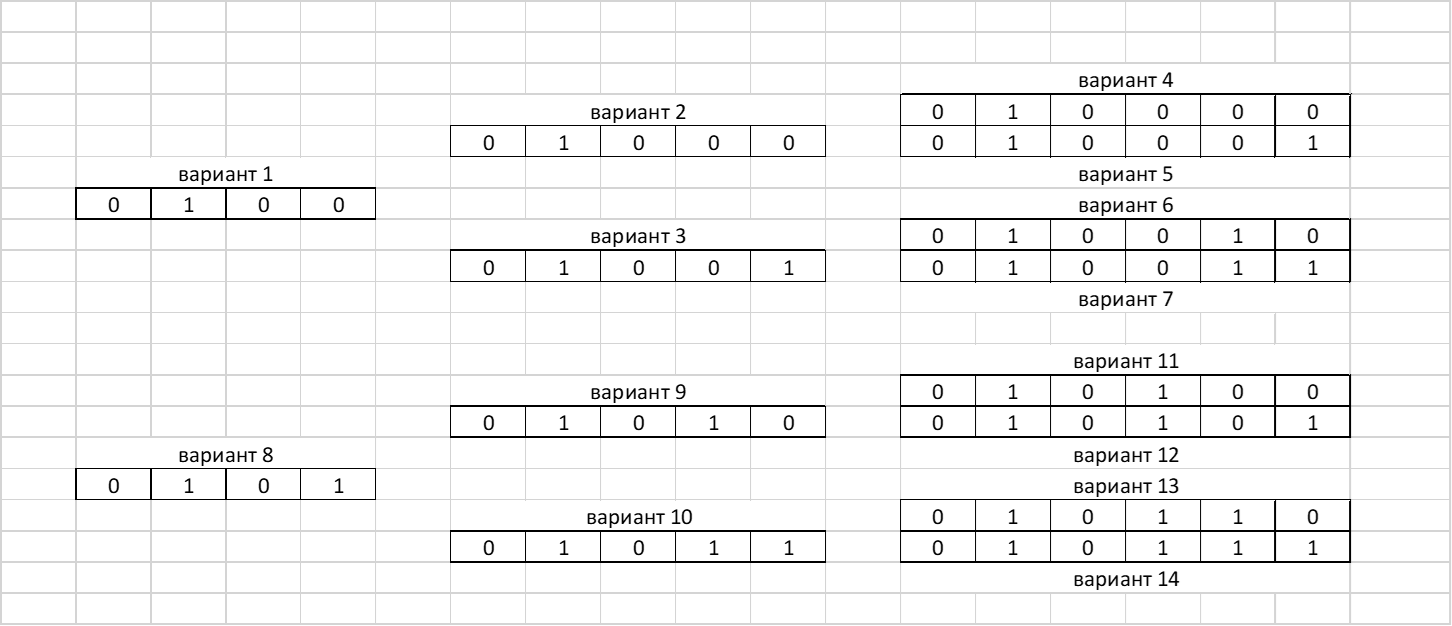
Отметим все кодовые цепочки для указанных букв, при этом нужно прервать все кодовые цепочки, которые они обрывают.

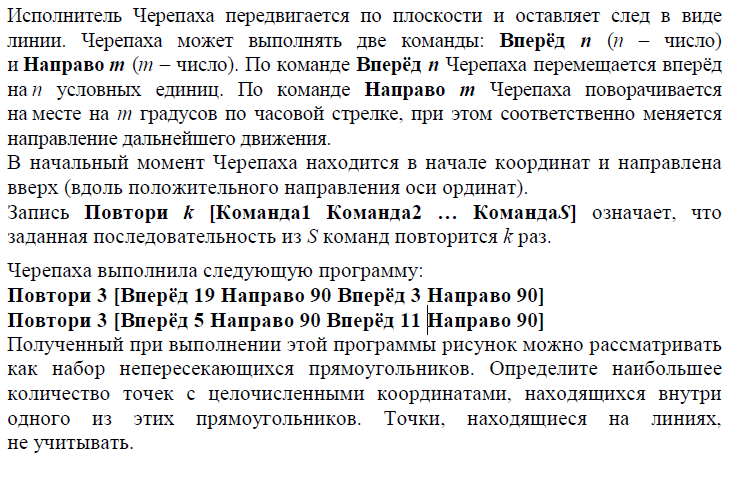


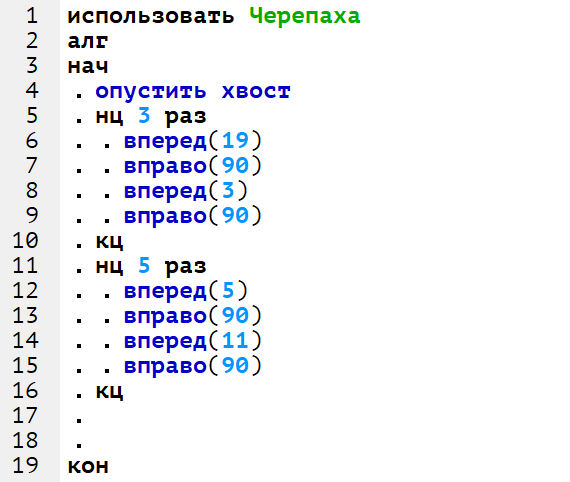
В результате получим следующую таблицу:

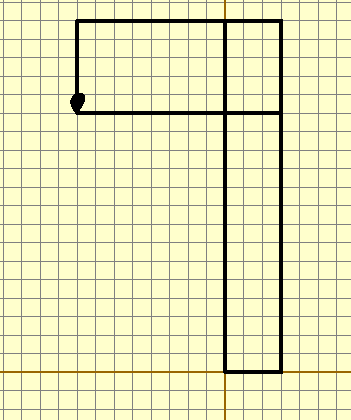


Код 010 использовать нельзя, так как не будет возможности построить кодовые цепочки для оставшихся букв. Тогда получается 14 всевозможных способов выбрать код для буквы Ю длинной от 4 до 6 знаков:

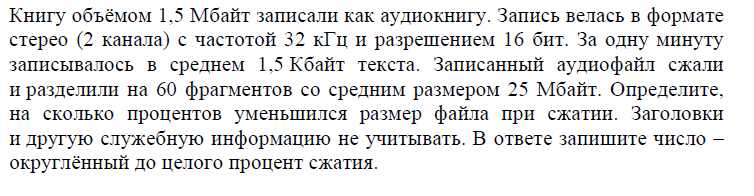












**Решение**

.

*(размер исходного файла)*

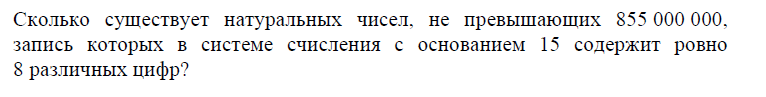
*(размер файла после сжатия)*

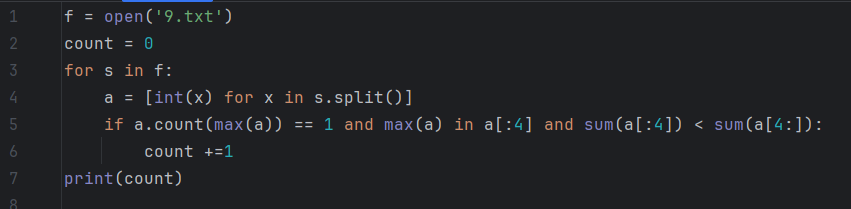
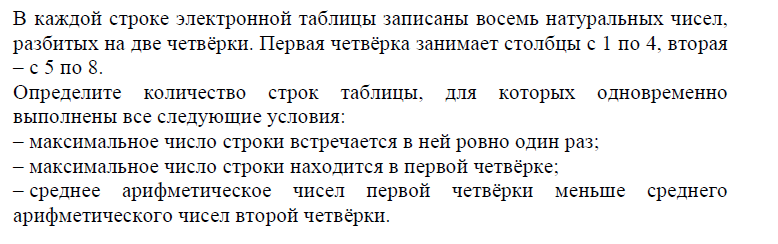
.

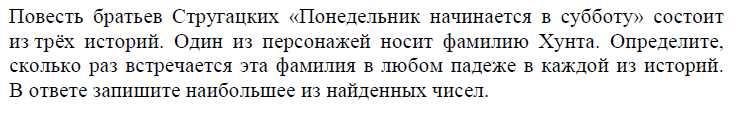
*.*

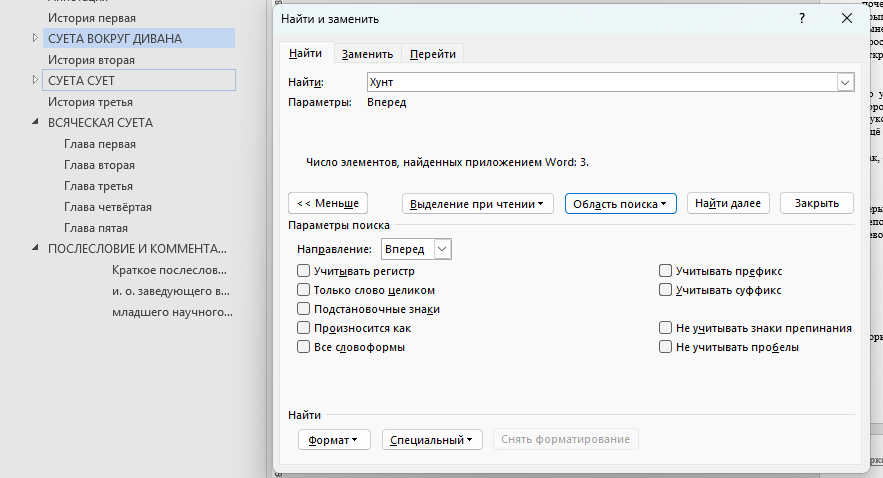
*(размер файла после сжатия)*

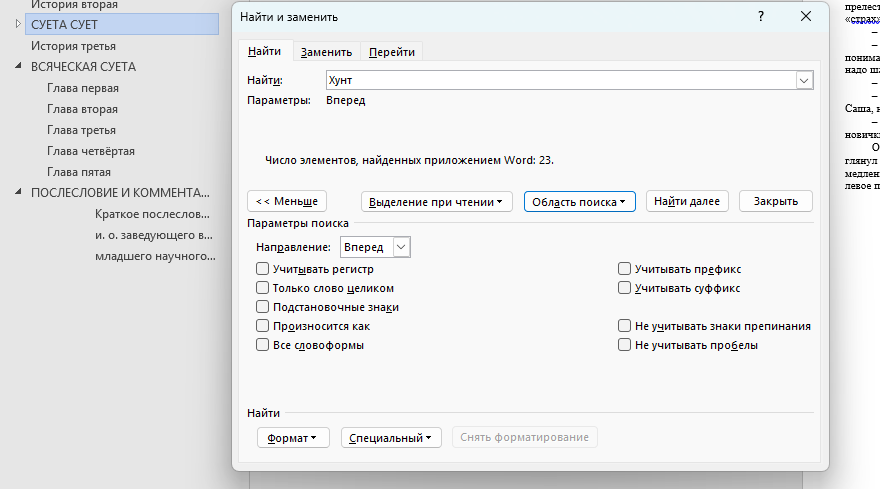
*.*

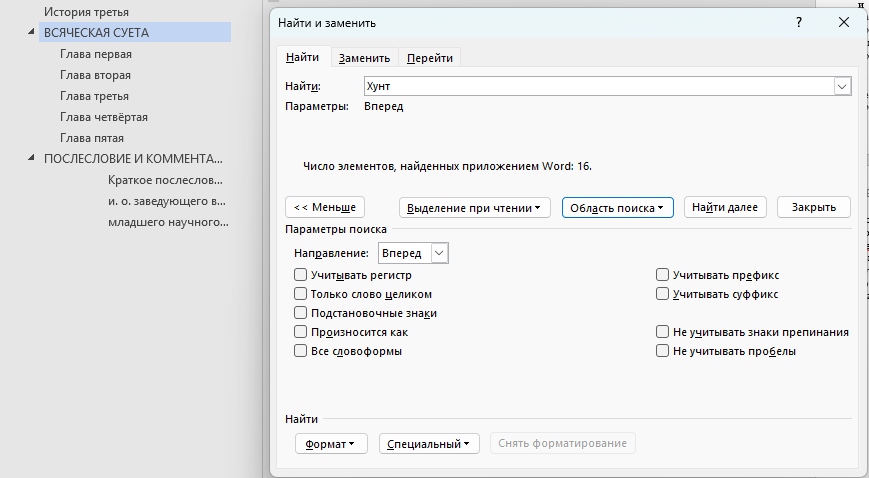


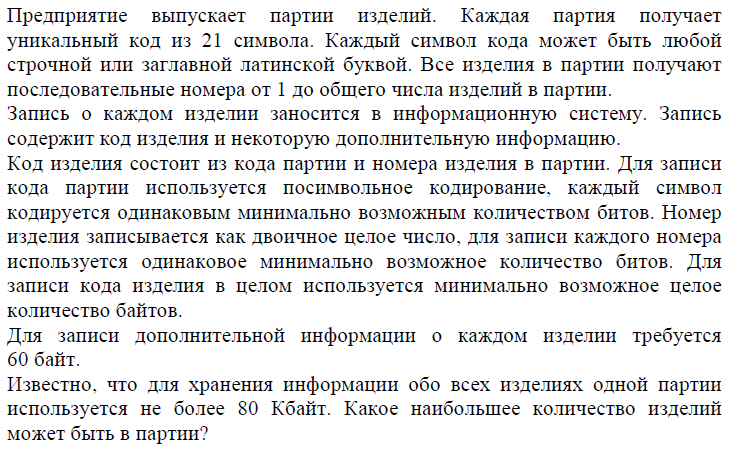




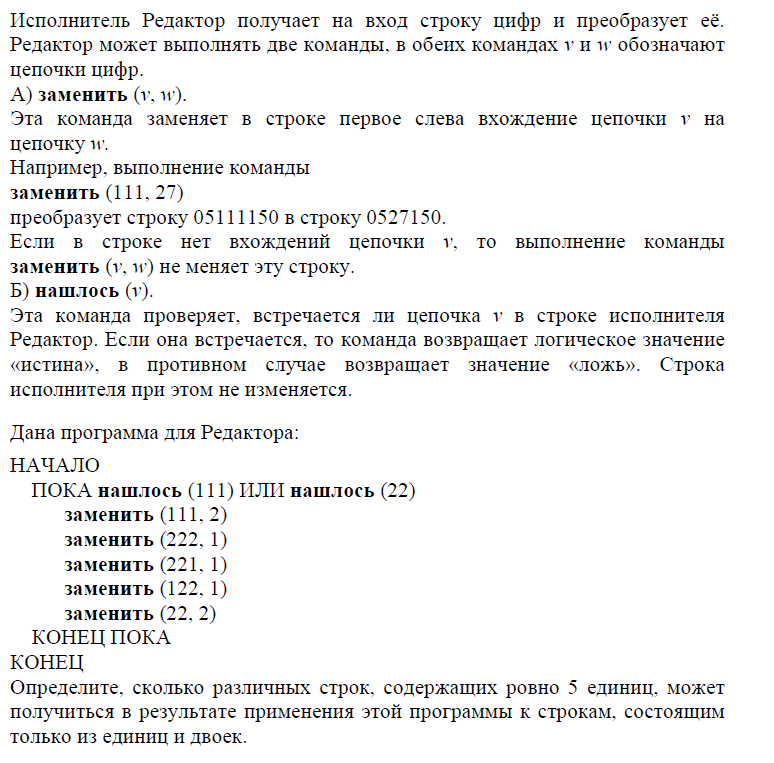


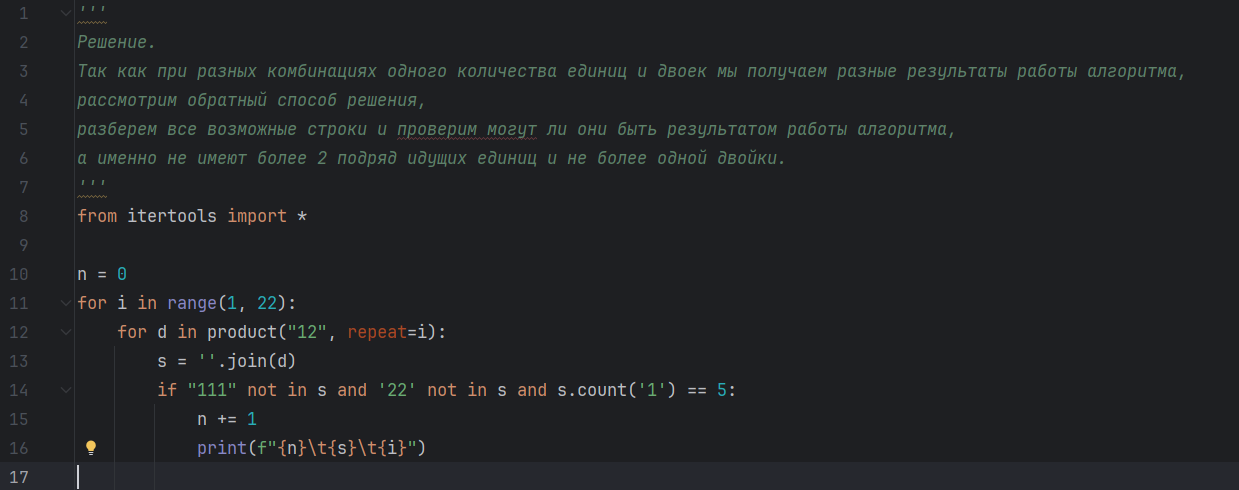


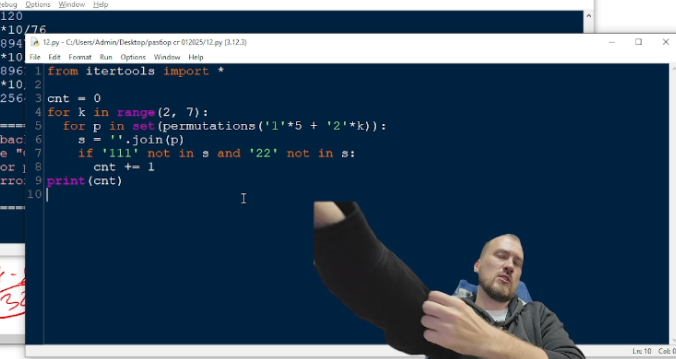


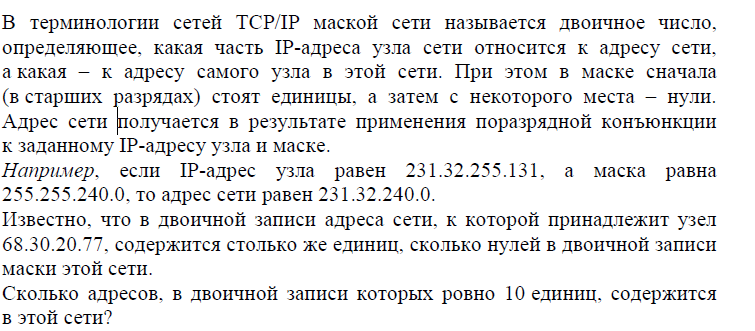


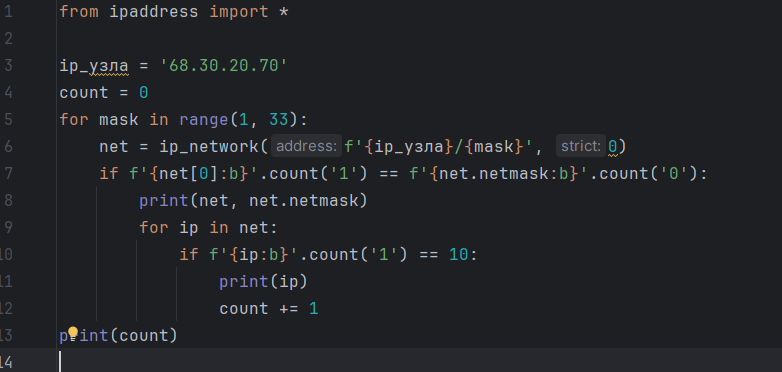






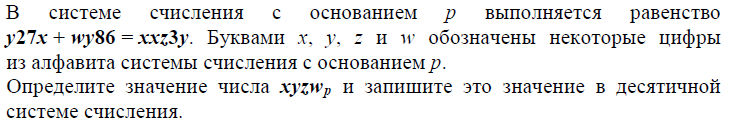


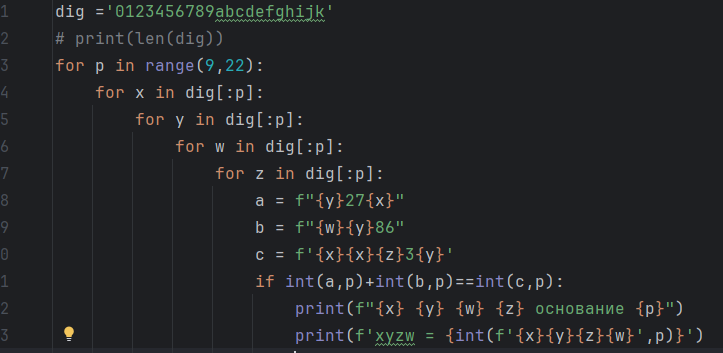


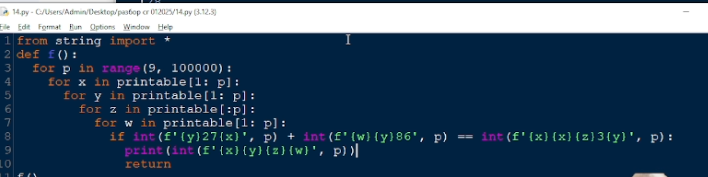


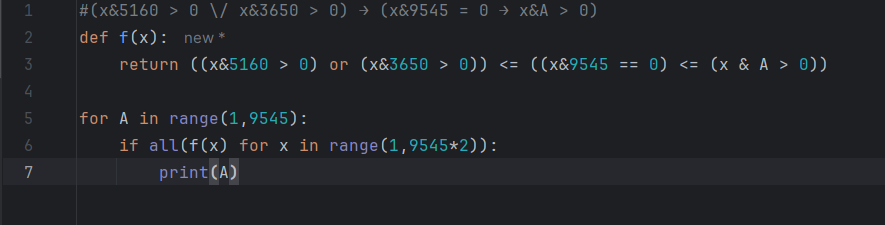
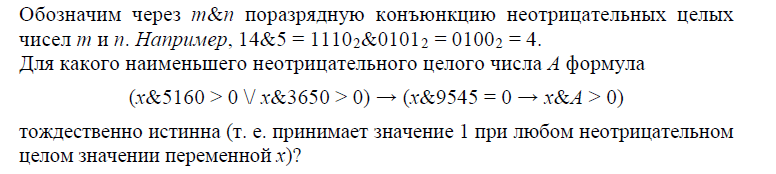
**Способ найти маску руками:**

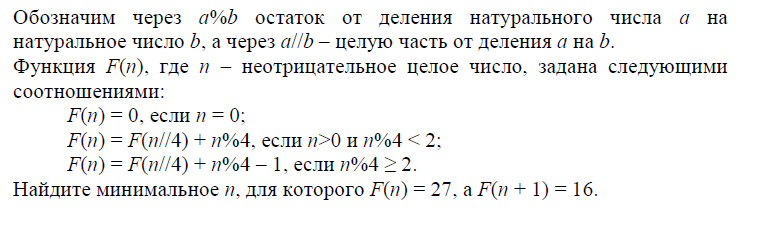
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **маска** | 11111111 | 11111111 | 11111111 | 00000000 |
| **IP-адрес узла** | 01000100 | 00011110 | 00010100 | 01001101 |
| **net** | 01000100 | 00011110 | 00010100 | 0000000 |
| **Net** | 68 | 30 | 20 | 0 |









.