**Лабораторная работа 4. Вариант 1.**

**Код Рида-Соломона.**

Выполнила студентка группы 09-941 Сарыймова Лейла.

**Параметры:**

Код Рида-Соломона - (14,10,4) (длина n кодового слова, количество k информационных и n-k проверочных символов, кодовое расстояние d)

Получен укорочением из кода - (15,11,5)

Примитивный многочлен – x4+x+1

**Ход работы:**

1. Определить примитивный элемент α конечного поля (r-разрядное слово)

α4 = α + 1

Поле GF(24):

|  |  |
| --- | --- |
| 𝛼0 | 0001 |
| 𝛼1 | 0010 |
| 𝛼2 | 0100 |
| 𝛼3 | 1000 |
| 𝛼4 | 0011 |
| 𝛼5 | 0110 |
| 𝛼6 | 1100 |
| 𝛼7 | 1011 |
| 𝛼8 | 0101 |
| 𝛼9 | 1010 |
| 𝛼10 | 0111 |
| 𝛼11 | 1110 |
| 𝛼12 | 1111 |
| 𝛼13 | 1101 |
| 𝛼14 | 1001 |

2. Задать порождающий многочлен g(x) степени n-k (как вектор из степеней примитивного элемента)

g(x)=α0\*x4+α13\*x3+α6\*x2+α3\*x1+α10

3. Построить порождающую и проверочную матрицы (вектора из степеней примитивного элемента), при необходимости, транспонированный вид

Порождающая матрица:

[0, '∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞']  
['∞', 0, '∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞']  
['∞', '∞', 0, '∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞']  
['∞', '∞', '∞', 0, '∞', '∞', '∞', '∞', '∞', '∞', '∞']  
['∞', '∞', '∞', '∞', 0, '∞', '∞', '∞', '∞', '∞', '∞']  
['∞', '∞', '∞', '∞', '∞', 0, '∞', '∞', '∞', '∞', '∞']  
['∞', '∞', '∞', '∞', '∞', '∞', 0, '∞', '∞', '∞', '∞']  
['∞', '∞', '∞', '∞', '∞', '∞', '∞', 0, '∞', '∞', '∞']  
['∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞', 0, '∞', '∞']  
['∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞', 0, '∞']  
['∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞', '∞', 0]  
[5, 13, 11, 6, 10, 13, 3, 10, 1, 1, 13]  
[3, 3, 10, 13, 14, 14, 12, 13, 11, 7, 6]   
[11, 7, 6, 3, 12, 9, 4, 13, 5, 8, 3]  
[8, 6, 1, 5, 8, 13, 5, 11, 11, 8, 10]

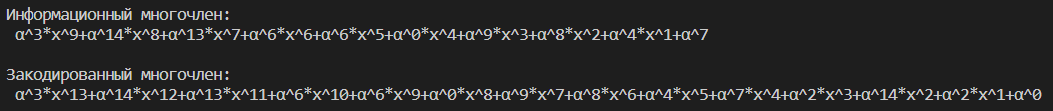
Проверочная матрица:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]  
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28]  
[0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42]  
[0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56]

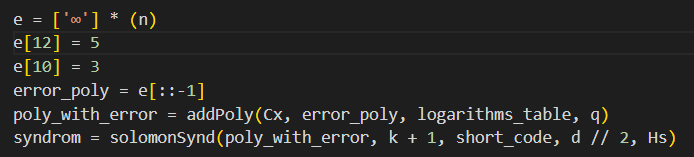
3. Выбрать информационное слово и закодировать его с помощью

порождающей матрицы.

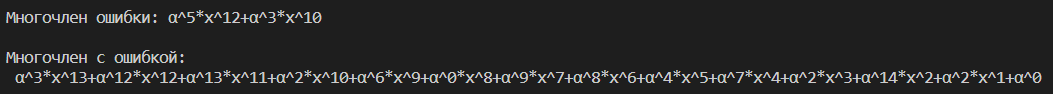
Зададим рандомный информационный многочлен и закодируем его.

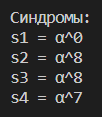


4. Внести в кодовое слово две ошибки и вычислить синдромы.

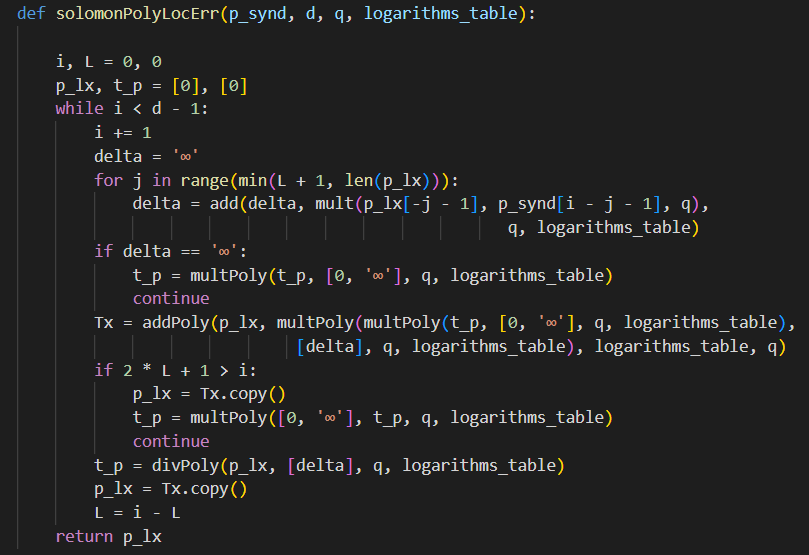


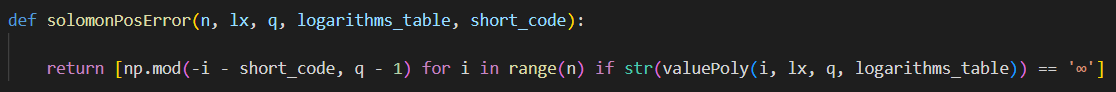
Внесли ошибки в 10 и 12 позиции. В результате получим:





5. Найти многочлен локаторов ошибки и вычислить корни (места ошибок).

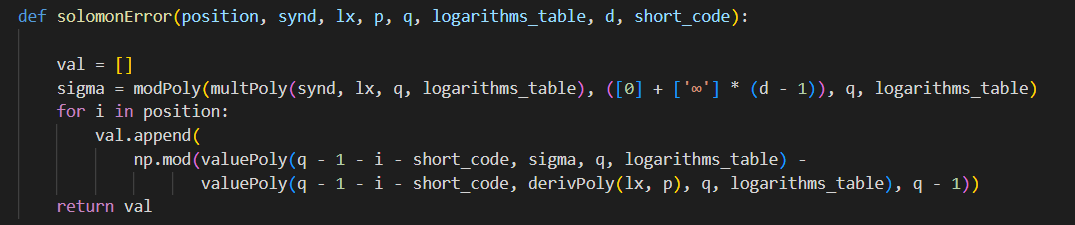








6. Найти величины ошибок.





ПРИЛОЖЕНИЕ

