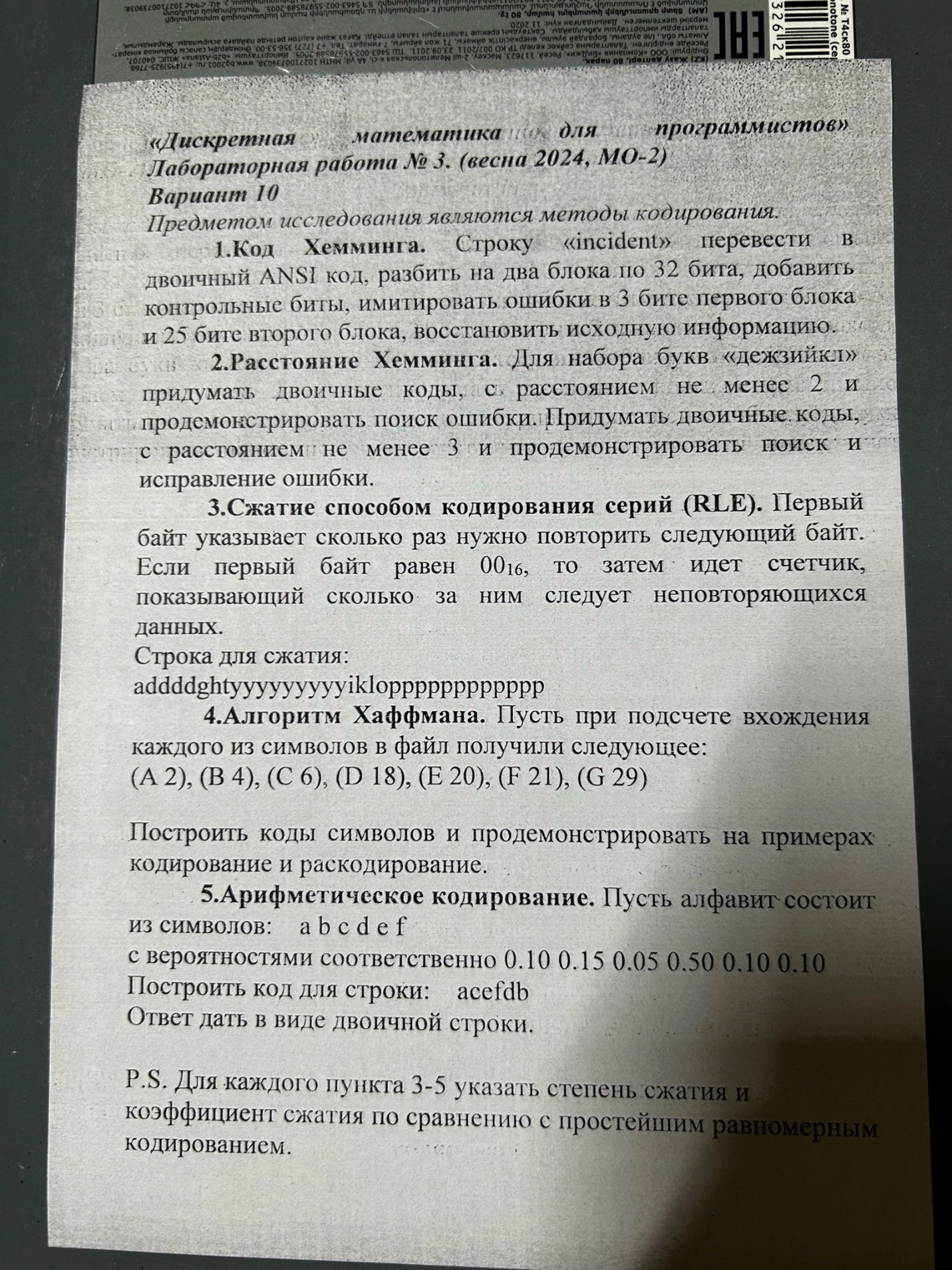
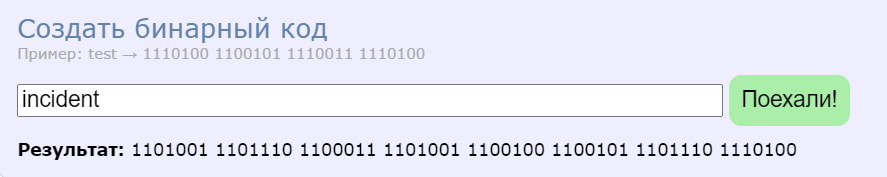
Лабораторная работа № 3

Алексенко Анна 2МОиАИС

Вариант 10



1. **Код Хемминга. Строку «incident» перевести в двоичный ANSI код, разбить на два блока по 32 бита, добавить контрольные биты, имитировать ошибки в 3 бите первого блока и 25 бите второго блока, восстановить исходную информацию**

Строка «» в бинарном виде, разбитая на два блока по 32 бита:

Кодировать блоки будем независимо. Вставим контрольные биты. Так как длина информационного слова бита, вставим их в позиции с номерами . Соответственно, имеем контрольных битов (они выделены красным цветом и обозначены символом «\*»):

Вычислим эти контрольные биты. Составим для каждого блока таблицу. Для первого и второго блоков имеем: [(см.excel)](C:\\Users\\aleks\\OneDrive\\Рабочий стол\\Institute\\Дискретная математика\\Лаба 3\\лаба3" \l "1(контрольные биты).xlsx)

В итоге получаем:

Имитация ошибок

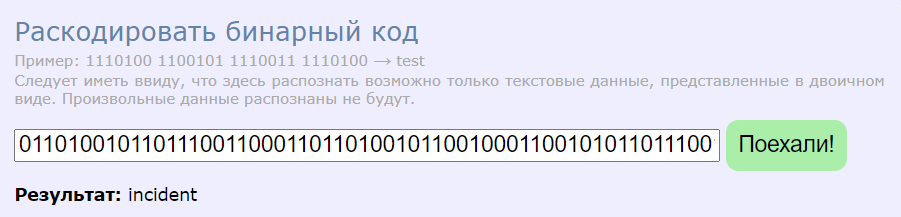
Выделим зелёным цветом изменённые бит первого блока и бит второго блока:

Вычислим заново все контрольные биты. Получаем, что не совпадают контрольные биты под номерами , в первом блоке и контрольные биты под номерами и во втором блоке. [(см.excel2)](file:///C:\Users\aleks\OneDrive\Рабочий%20стол\Institute\Дискретная%20математика\Лаба%203\лаба3#1(имитация ошибки).xlsx)

Суммы номеров позиций неправильных контрольных битов равны позициям ошибочных битов:

Инвертировав изменённые биты и отбросив контрольные биты, получим исходное сообщение в первозданном виде. Инвертируем биты:

Отбрасываем контрольные:



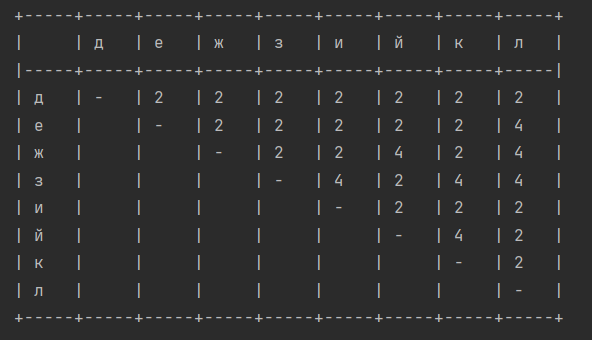
**2) Расстояние Хемминга. Для набора букв «дежзийкл» придумать двоичные коды с расстоянием не менее 2 и продемонстрировать поиск ошибки. Придумать двоичные коды с расстоянием не менее 3 и продемонстрировать поиск и исправление ошибки**

Двоичные коды с расстоянием не менее 2

Таблица двоичных кодов:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| д |  | з |  | к |  |
| е |  | и |  | л |  |
| ж |  | й |  |

Таблица попарных расстояний: [(см.код)](file:///C:\Users\aleks\OneDrive\Рабочий%20стол\Institute\Дискретная%20математика\Лаба%203\code\#2main.py)



Декодер исправляет все ошибки, кратность которых не превышает



Минимальное расстояние => код может исправить ноль ошибок, обнаружить одну ошибку

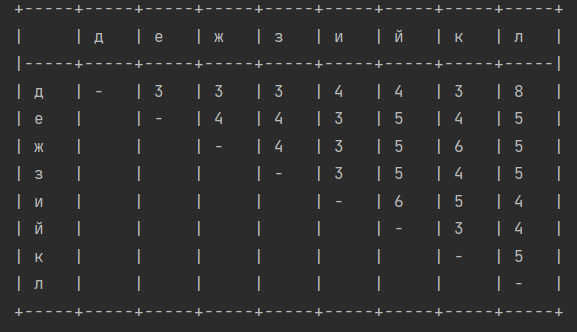
Пусть было получено сообщение 00100. Совпадений кода с кодами в таблице двоичных кодов нет. Посчитаем расстояния Хемминга для полученного кода и исходного набора букв:

Минимальное расстояние соответствует трём символам. Найдена одиночная ошибка, исправление невозможно

Двоичные коды с расстоянием не менее 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| д | 00000000 | з | 11000001 | к | 10100100 |
| е | 00000111 | и | 01010101 | л | 11111111 |
| ж | 00011001 | й | 10110010 |

Таблица попарных расстояний: [(см.код2)](https://d.docs.live.net/a22d9a578a4f0f20/Рабочий%20стол/Institute/Дискретная%20математика/Лаба%203/code/%5eN2/main2.py)



Минимальное расстояние , значит такой код может исправить единичную ошибку, обнаружить две ошибки

Пусть было получено сообщение . Совпадений с кодами в таблице двоичных кодов не обнаружено. Посчитаем расстояния Хемминга для полученного кода и исходного набора букв:

Минимальное расстояние соответствует символу «». Найдена и исправлена одиночная ошибка

**3) Сжатие способом кодирования серий (RLE). Первый байт указывает сколько раз нужно повторить следующий байт. Если первый байт равен 0016, то затем идет счетчик, показывающий сколько за ним следует неповторяющихся данных. Строка для сжатия: аddddghtyyyyyyyyyiklopppppppppppp**

аddddghtyyyyyyyyyiklopppppppppppp (33 байт)

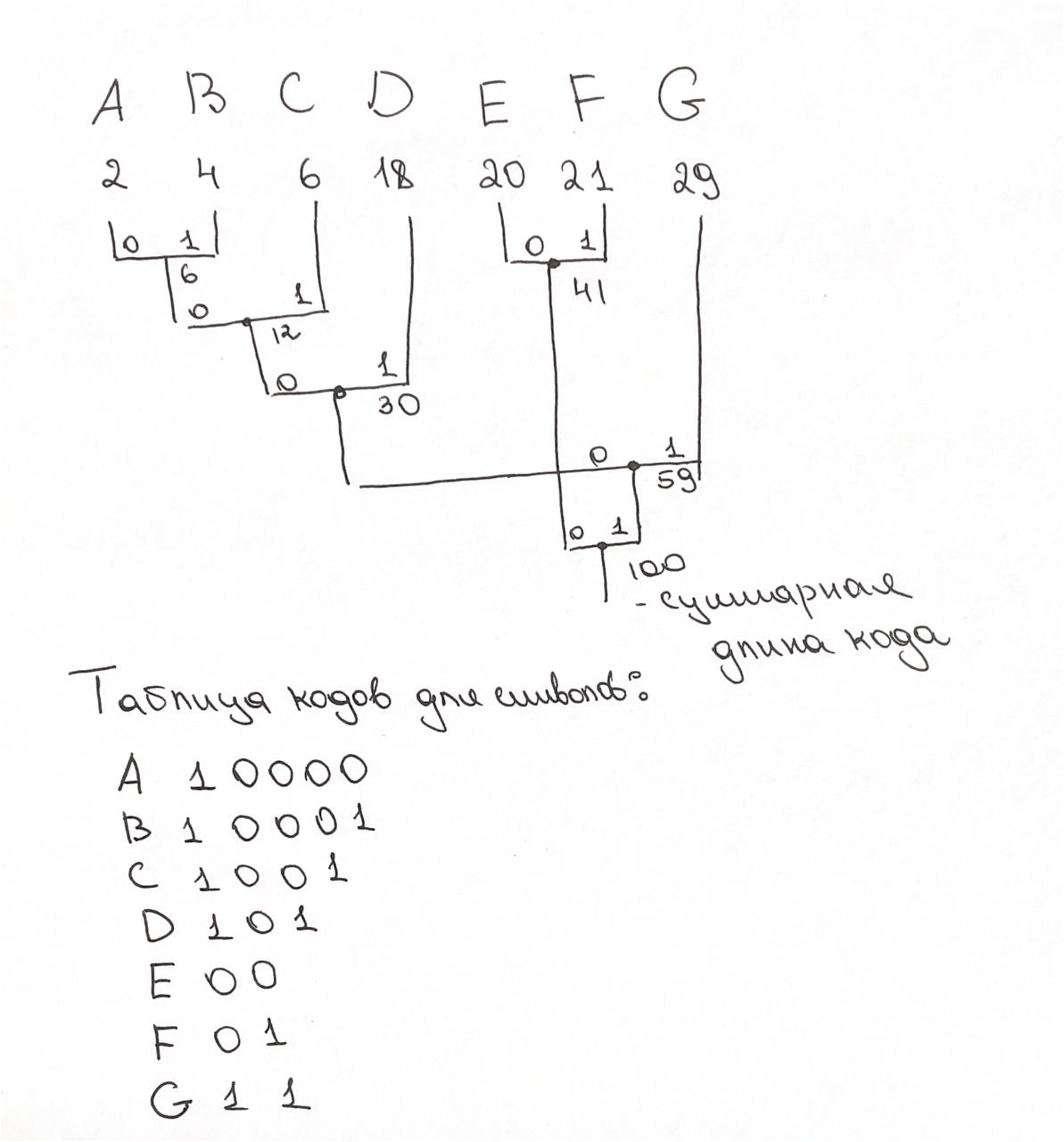
1a4d03ght9y04iklo12p ( 19 байт)

Степень сжатия

Коэффициент сжатия

**4) Алгоритм Хаффмана. Пусть при подсчете вхождения каждого из символов в файл получили следующее:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Символ** | **A** | **B** | **C** | **D** | **E** | **F** | **G** |
| **Число вхождений** | **2** | **4** | **6** | **18** | **20** | **21** | **29** |

**Построить коды символов и продемонстрировать на примерах кодирование и раскодирование**

Например**,** закодируем слово CAGE с помощью таблицы:

1001100001100

Декодируем с помощью дерева

При равномерном кодировании каждый символ кодируется

Посчитаем среднюю длину:

Степень сжатия

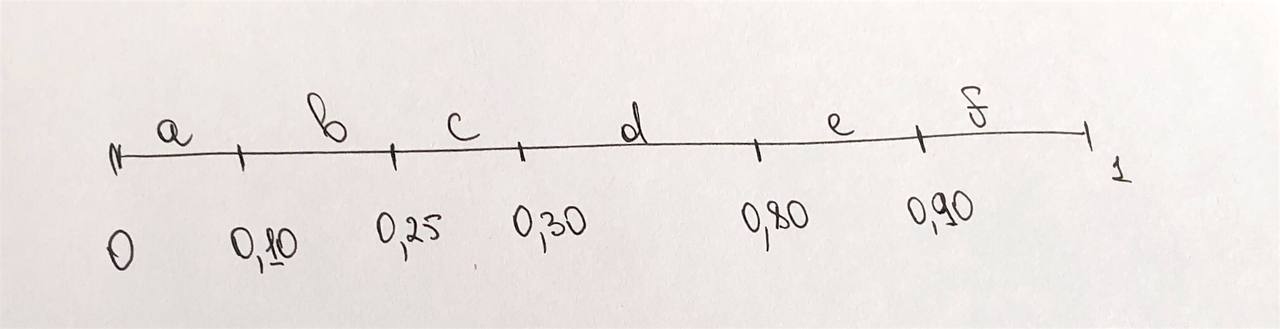
Коэффициент сжатия

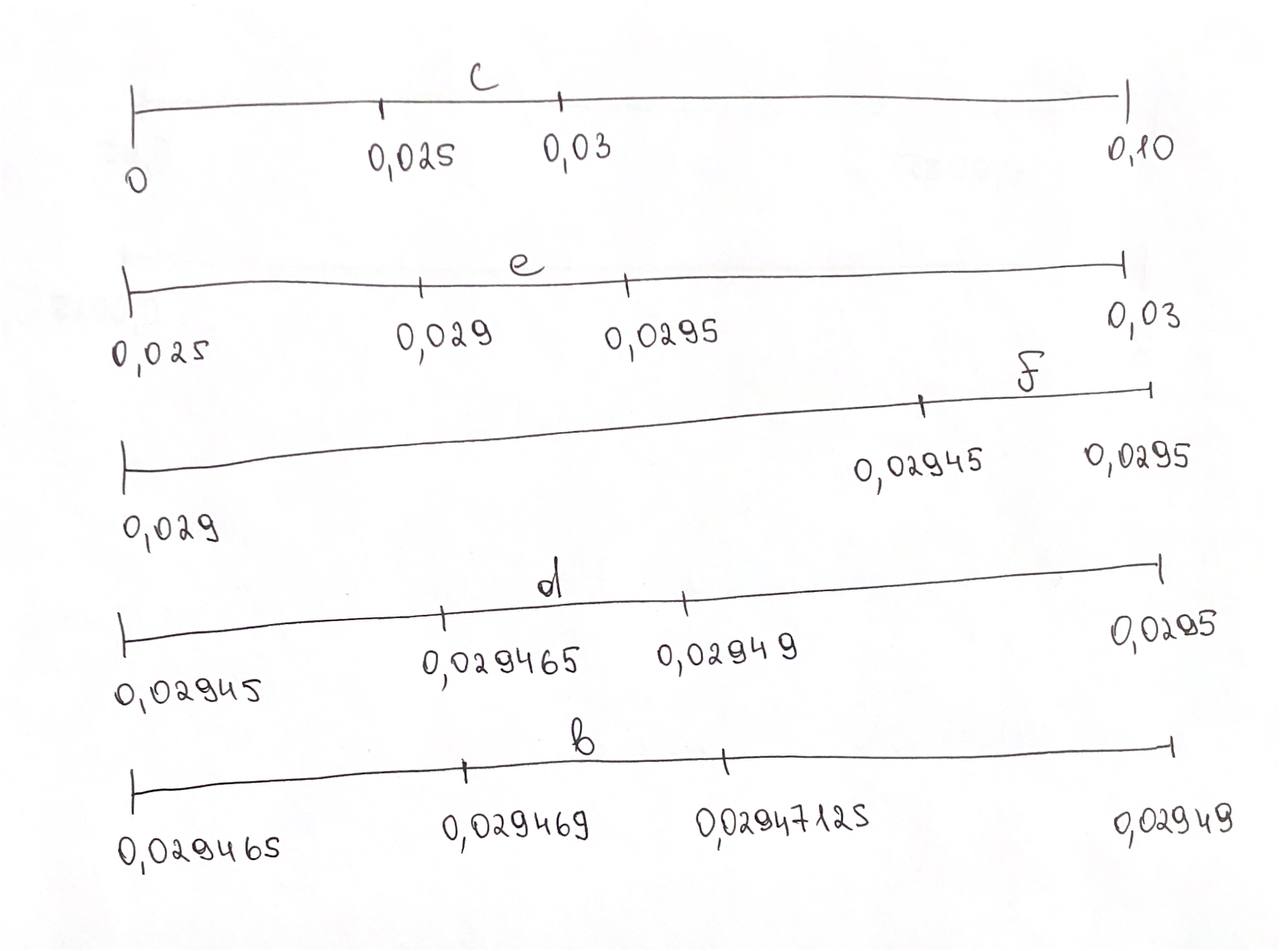
**5) Арифметическое кодирование.**

**Пусть алфавит состоит из символов: a b c d e f**

**с вероятностями соответственно 0,10 0,15 0,05 0,50 0,10 0,10**

**Построить код для строки: acefdb. Ответ дать в виде двоичной строки**





Acefdb – отрезок

Например,

В виде двоичной строки: 111001100011101