МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ (ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ МАШИН СИСТЕМ И СЕТЕЙ

Лабораторная работа №1 по курсу «Методы и средства передачи информации»

Тема: «Количество информации. Энтропия»

Задачи

Задание №1.

Составить программу для определения какое количество информации несет сообщение о том, что горит красный и желтый сигнал светофора, среднее количество информации о том, что загорелся любой из трех сигналов.

Задание №2.

Для кодирования строки равномерным двоичным кодом, например, для слова «машина», алфавит: м,а,ш,н,и – 5 букв, достаточно 3 бита.

Кодовый словарь:

M 001

A 010

Ш 011

И 100

H 101

Слово «машина» кодируется сообщением 001010011100101010.

Составить программу, которая кодирует строку любой длины, составляет кодовый словарь и выводит закодированное сообщение.

Описание результатов

Задание №1

 $i_x = \log_2(\frac{1}{p_x})$ – количество информации при наступлении события х

 $p_x = \frac{k}{n}, k$ — количество событий x, n — количество всех событий — вероятность наступления события x

$$H = \sum_{x=1}^n p_x \log_2(\frac{1}{p_x})$$
 – среднее количество информации

Результатом работы программы будет применение этих формул к заданным данным (длительности горения каждого сигнала светофора).

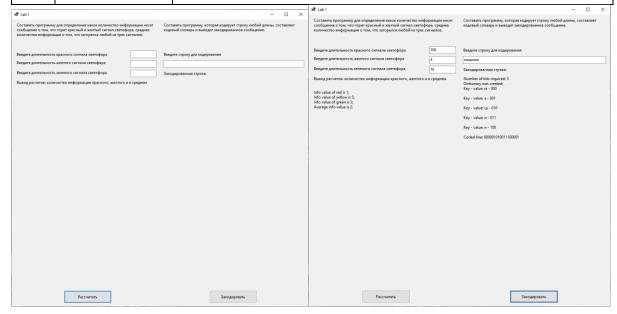
№ теста	Входные данные (красный, желтый, зеленый)	Выходные данные (информация красного, желтого, зеленого, в среднем)
1	108 4 16	1532
2	000	Не требуется нисколько количества информации
3	10 10 10	2222

Задание №2

Результатом работы программы станет количество необходимых бит для кодировки, словарь кодировки и закодированная строка. От количества бит зависит длина двоичного кода каждого символа.

Табл.1 Тесты задания 2

№ тест а	Входные данные	Выходные данные (количество бит, словарь, закодированная строка)
1	машина	3 (м – 000) (а - 001) (ш – 010) (и - 011) (н – 100) 000001010011100001
2	шоссе	3 (ш - 000) (о - 001) (с - 010) (е - 011) 000001010010011
3	корпускулярно сть	4 (к - 0000) (о - 0001) (р - 0010) (п - 0011) (у - 0100) (с - 0101) (л - 0110) (я - 0111) (н - 1000) (т - 1001) (ь - 1010) 0000000100100011010000101000010011100101



Текст программы (основное)

```
public Part1(int red duration, int yellow duration, int green duration)
        red = red duration;
        yellow = yellow_duration;
green = green_duration;
        // Создаю переменную, отвечающую за общее время цикла светофора
        sum = red + yellow + green;
        // Создаю переменные, отвечающие за вероятности происшествия каждого сигнала свето\phiора
        pRed = red / sum;
        pYellow = yellow / sum;
        pGreen = green / sum;
        // Создаю переменные, отвечающие за количество информации, которую несет каждый сигнал
        // int iRed = (int) Math.Ceiling(((Math.Log(1 / pRed) / Math.Log(2))));
        iRed = (int) Math.Ceiling(Math.Log2(1 / pRed));
        // int iYellow = (int) Math.Ceiling(((Math.Log(1 / pYellow) / Math.Log(2))));
        iYellow = (int) Math.Ceiling(Math.Log2(1 / pYellow));
        // int iGreen = (int) Math.Ceiling(((Math.Log(1 / pGreen) / Math.Log(2))));
        iGreen = (int) Math.Ceiling(Math.Log2(1 / pGreen));
        // Создаю переменную, отвечающую за среднее значение информации, которую несут сигналы
светофора
        avg = Math.Ceiling(pRed * iRed + pYellow * iYellow + pGreen * iGreen);
    public Part2(string input line)
        dictionary = new Dictionary<char, string>();
        counter = 0;
        foreach (char element in input line)
            if (!dictionary.ContainsKey(element))
                counter++;
            }
        bits = (int) Math.Ceiling(Math.Log(counter) / Math.Log(2));
        foreach (char element in input line)
            if (!dictionary.ContainsKey(element))
                string meaning = Convert.ToString(value, 2);
                meaning = meaning.PadLeft(bits, '0');
                dictionary.Add(element, meaning);
                value++;
        output = null;
        foreach (char symbol in input_line)
            output += dictionary[symbol];
```

Выводы

Задание 1: Зная количество всех событий и нужного события, можно найти количество информации, необходимое для его передачи и среднее количество информации для каждого события.

Задание 2: Рассчитав количество битов, необходимое для кодировки информации, можно ограничить количество используемой памяти и ускорить работу программы.