

Министерство образования Республики Беларусь
Учреждение образования
“Брестский государственный технический университет”
Кафедра интеллектуально-информационных технологий

Лабораторная работа №1
“Код Хемминга”

По дисциплине “Современные методы защиты компьютерных систем”

Выполнила:
студентка 4 курса
группы ИИ-22
Леваневская Н.И.
Проверил:
Хацкевич А.С.

Цель работы: реализовать код Хемминга для своего варианта.

Вариант 11

Ход работы:

Задание. Составить код Хемминга (классический алгоритм) ($M+r$, M), допустить ошибку в одном из разрядов и отыскать её по алгоритму. $M = 590$, $r = 5$.

Код программы:

```
using System;

class HammingCode
{
    public static int[] ConvertToBinary(int number)
    {
        string binaryString = Convert.ToString(number, 2);
        int[] binaryArray = new int[binaryString.Length];
        for (int i = 0; i < binaryString.Length; i++)
        {
            binaryArray[i] = int.Parse(binaryString[i].ToString());
        }
        return binaryArray;
    }

    public static int[] EncodeHamming(int[] data)
    {
        int[] encoded = new int[16];

        encoded[2] = data[0];
        encoded[4] = data[1];
        encoded[5] = data[2];
        encoded[6] = data[3];
        encoded[8] = data[4];
        encoded[9] = data[5];
        encoded[10] = data[6];
        encoded[11] = data[7];
        encoded[12] = data[8];
        encoded[13] = data[9];

        encoded[0] = encoded[2] ^ encoded[4] ^ encoded[6] ^ encoded[8] ^ encoded[10] ^ encoded[12];
        encoded[1] = encoded[2] ^ encoded[5] ^ encoded[6] ^ encoded[9] ^ encoded[10] ^ encoded[13];
        encoded[3] = encoded[4] ^ encoded[5] ^ encoded[6] ^ encoded[11] ^ encoded[12] ^ encoded[13];
        encoded[7] = encoded[8] ^ encoded[9] ^ encoded[10] ^ encoded[11] ^ encoded[12] ^
encoded[13];
        encoded[15] = encoded[0] ^ encoded[1] ^ encoded[3] ^ encoded[7];

        return encoded;
    }

    public static void DetectAndCorrectError(int[] encoded)
    {
        int p1 = encoded[0] ^ encoded[2] ^ encoded[4] ^ encoded[6] ^ encoded[8] ^ encoded[10] ^
encoded[12];
        int p2 = encoded[1] ^ encoded[2] ^ encoded[5] ^ encoded[6] ^ encoded[9] ^ encoded[10] ^
encoded[13];
        int p4 = encoded[3] ^ encoded[4] ^ encoded[5] ^ encoded[6] ^ encoded[11] ^ encoded[12] ^
encoded[13];
        int p8 = encoded[7] ^ encoded[8] ^ encoded[9] ^ encoded[10] ^ encoded[11] ^ encoded[12] ^
encoded[13];
        int p16 = encoded[15] ^ encoded[0] ^ encoded[1] ^ encoded[3] ^ encoded[7];

        int errorPosition = p1 + (p2 * 2) + (p4 * 4) + (p8 * 8) + (p16 * 16);

        if (errorPosition != 0)
```

```

        {
            Console.WriteLine($"Ошибка обнаружена в бите {errorPosition}, исправляем...");
            encoded[errorPosition - 1] ^= 1;
        }
        else
        {
            Console.WriteLine("Ошибок нет.");
        }
    }

    static void Main(string[] args)
    {
        int number = 590;
        int[] data = ConvertToBinary(number);
        Console.WriteLine("Исходные данные: " + string.Join("", data));

        int[] encoded = EncodeHamming(data);
        Console.WriteLine("Закодированные данные: " + string.Join("", encoded));

        encoded[4] ^= 1;
        Console.WriteLine("Данные с ошибкой: " + string.Join("", encoded));

        DetectAndCorrectError(encoded);
        Console.WriteLine("Исправленные данные: " + string.Join("", encoded));
    }
}

```

Результат работы программы:

```

Исходные данные: 1001001110
Закодированные данные: 0111001100111001
Данные с ошибкой: 0111101100111001
Ошибка обнаружена в бите 5, исправляем...
Исправленные данные: 0111001100111001

```

Вывод: Изучила и реализовала код Хемминга.