Учреждение образования

«Белорусский государственный технологический университет»

**Защита информации и надежность информационных систем**

**Лабораторная работа №7**

ПЕРЕМЕЖЕНИЕ/ДЕПЕРЕМЕЖЕНИЕ ДАННЫХ В ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Выполнил:

Студент 3 курса 2 группы ФИТ

Максимова Вера Владимировна

Проверил:

Ржеутская Надежда Викентьевна

**2022 г.**

Цель: приобретение практических навыков использования методов перемежения/деперемежения двоичных данных в информационных системах. Задачи: 1. Закрепить теоретические знания по алгебраическому описанию и использованию методов перемежения/деперемежения двоичных данных в информационных системах. 2. Разработать приложение для реализации метода перемежения/деперемежения символов в сообщениях на основе двоичного алфавита. 3. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

namespace Lab7

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("<---Передача информации с использованием кода Хемминга и блокового перемежителя--->");

Console.WriteLine($"\nПоток поделен на информационные слова: \nдлина потока - 16\nдлина информационного слова - 4\n");

int flowLenght = 16;

int k = (int)(Math.Sqrt(flowLenght));

int r = HemmingCode.GetNumOfVerifChar(k);

int n = k + r;

int codeCombinationLenght = flowLenght + (r \* k);

int[] informationFlow = new int[16] { 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1};

int[] infoFlowAfterChanges = new int[flowLenght];

int[] codeCombination = new int[flowLenght + (r \* k)];

int[,] HemmingsVerificMatrix = new int[n, r];

Console.WriteLine("Структура информационного потока на входе кодера: ");

MatrixOperations.PrintArray(informationFlow);

HemmingsVerificMatrix = HemmingCode.GetVerifMatrix(k);

//MatrixOperations.PrintVerifMatrix(HemmingsVerificMatrix, n, r);

Interleaver.AddCheckBits(informationFlow, codeCombination, HemmingsVerificMatrix);

Console.WriteLine("\n\nПоследовательность символов сообщения на выходе кодера (после добавления проверочных битов): ");

MatrixOperations.PrintArray(codeCombination);

Interleaver.ExecInterleaving(codeCombination, k);

Console.WriteLine("\nЗакодированное сообщение после перемежения: ");

MatrixOperations.PrintArray(codeCombination);

try

{

Console.WriteLine("\n\n<---Предположим, что в процессе передачи информации по каналу возник пакет ошибок--->");

Console.WriteLine("Укажите позицию начала пакета ошибки: ");

int error = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Укажите длину пакета ошибок (3, 5, 7): ");

int errorLenght = Convert.ToInt32(Console.ReadLine());

if (errorLenght == 3 || errorLenght == 5 || errorLenght == 7)

{

for (int i = error; i < (error + errorLenght); i++)

{

codeCombination[i] = (codeCombination[i] + 1) % 2;

}

Console.WriteLine("\nПередаваемое сообщение, содержащее группу ошибок: ");

MatrixOperations.PrintArray(codeCombination);

Interleaver.ExecDeInterleaving(codeCombination, k);

Console.WriteLine("\nСобщение с разнесёнными ошибками: ");

MatrixOperations.PrintArray(codeCombination);

Interleaver.LineByLineSearchError(codeCombination, HemmingsVerificMatrix, k);

Console.WriteLine("\n\nИнформационный поток на выходе декодера кода Хемминга (после исправления ошибок): ");

MatrixOperations.PrintArray(codeCombination);

Interleaver.RemoveCheckBits(infoFlowAfterChanges, codeCombination, HemmingsVerificMatrix);

Console.WriteLine("\n\nИнформационный поток после удаления проверочных бит: ");

MatrixOperations.PrintArray(infoFlowAfterChanges);

Console.WriteLine("\n\nИсходный информационный поток: ");

MatrixOperations.PrintArray(informationFlow);

Console.WriteLine();

}

else

{

Console.WriteLine("!Для выполнения задания по варианту необходимо ввести длину пакета ошибок равную 3, 5 или 7!");

}

}

catch

{

Console.WriteLine("!Вероятно допущена ошибка при указании позиции начала пакета ошибок, либо длины пакета!");

}

}

}

}