Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



**Отчёт**

**По домашнему заданию №2**

**По курсу «Сети и телекоммуникации»**

**Вариант 18**

**ИСПОЛНИТЕЛЬ:**

Группа ИУ5-55Б

Шакиров Т.М.

"7" ноября 2024 г.

**ПРЕПОДАВАТЕЛЬ:**

Галкин В.А.

"7" ноября 2024 г.

Москва, 2024

**Постановка и метод решения задачи для варианта задания.**

Имеется дискретный канал связи, на вход которого подается закодированная в соответствии с вариантом задания кодовая последовательность. В канале возможны ошибки любой кратности. Вектор ошибки может принимать значения от единицы в младшем разряде до единицы во всех разрядах кодового вектора. Для каждого значения вектора ошибки на выходе канала после декодирования определяется факт наличия ошибки.

**Вариант 18**

Необходимо использоватькодирование циклическим кодом Ц [7, 4], а также следует определить обнаруживающую способность кода С0:

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант  № | Информационный вектор | Код | Способность кода |
| 18 | 1000 | Ц[7,4] | C0 |

**Реализация модели канала связи, алгоритмы кодирования, декодирования и вычисления обнаруживающей способности кода для ошибок всех возможных кратностей:**

Задача была выполнена с использованием языка Python. Для табличной формы необходим модуль ‘tabulate’.

**Код программы:**

import math as math

from typing import Dict

from tabulate import tabulate

perevod: Dict[int, int] = {1:0b1, 2: 0b10, 4: 0b100, 3:0b1000, 6:0b10000,

5:0b1000000, 7: 0b100000}

def bin\_to\_dec(string):

    return int(string, 2)

def deller(delymoe):

    delytel = bin\_to\_dec("1011")

    a = True

    difference = 1

    while(difference >= 0):

        difference = bin(delymoe).\_\_len\_\_() - bin(delytel).\_\_len\_\_()

        if difference>0:

            delymoe = delymoe ^ (delytel<<difference)

        else:

             delymoe = delymoe ^ delytel

        difference = bin(delymoe).\_\_len\_\_() - bin(delytel).\_\_len\_\_()

    return delymoe

def deller1(a):

    b=11

def workwitherror(countoferrors, error):

    error += 1

    a = True

    while a:

        str = bin(error)

        countofone = 0

        for i in str:

            if i == '1':

                countofone += 1

        if countofone == countoferrors:

            break

        else:

            error +=1

    return error

def main():

    for i in range(1,1000):

        if(deller(i)!=deller1(i)):

            print("Ошибка\n")

def main1():

    a = str(input('Введите вектор:'))

    b = bin\_to\_dec(a)

    b = b << 3

    rez = ''

    ostatok = deller(b)

    b = b | ostatok

    table = []

    for i in range(1,8):

        error = 0b0

        Countisprav = 0

        CountNaiden = 0

        combinations = int(math.factorial(7)/ (math.factorial(i)\*

math.factorial(7-i)))

        for j in range (1, combinations +1):

            vectorbuff = b

            error = workwitherror(i, error)

            vectorbuff = vectorbuff ^ error

            vectorsyndroma = deller(vectorbuff)

            if vectorsyndroma != 0:

                CountNaiden += 1

                errorbytable = perevod[vectorsyndroma]

                vectorbuff = vectorbuff ^ errorbytable

                if vectorbuff == b:

                    Countisprav+=1

        Co = CountNaiden/ combinations

        if Co == 1:

            rez = 'Все ошибки найдены'

        if Co == 0:

            rez = 'Ошибок не найдено'

        if Co != 0 and Co != 1:

          rez = 'Часть ошибок найдено'

        table.append([i, combinations, CountNaiden, Countisprav,

'{:.1%}'.format(Co ), rez])

    headers = ['Количество ошибок', 'Количество комбинаций','Количество найденных', 'Количество исправленных', 'Обнаруживающая способность', 'Результат проверки']

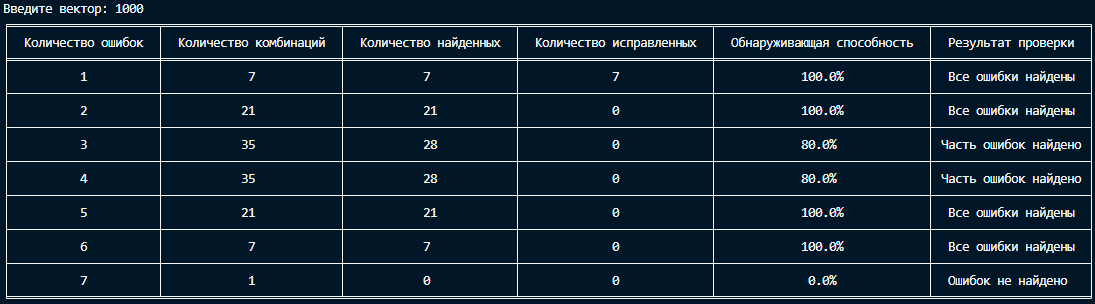
    print(tabulate(table, headers, tablefmt='fancy\_grid', stralign='center',

numalign='center'))

if \_\_name\_\_=='\_\_main\_\_':

    main1()

**Заполненная программно таблица результатов:**

****

**Выводы**

Проанализировав таблицу, видим, что при применении циклического кода Ц[7, 4] обнаруживающая способность равна 100.0% при кратности ошибок 1,2,5,6, и 0.0% при кратности 7. При кратности ошибок 3 и 4 - 80%.

**Список используемой литературы и URL-ссылок.**

* Галкин В.А. Методическое пособие по выполнению домашнего задания по дисциплине «Сети и телекоммуникации», 2018
* Галкин В.А., Григорьев Ю.А. Телекоммуникации и сети: Учеб. Пособие для вузов.- М.: Изд-во МГТУ им.Н.Э.Баумана, 2003
* Библиотека Tabulate для Python. <https://pypi.org/project/tabulate/>
* Справочная информация по языку Python. <https://www.jetbrains.com/help/pycharm/2020.2>