

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ
по лабораторной работе №3
на тему

Методы защиты от ошибок

Выполнил студент группы № 050503

Казак И. А

Преподаватель

Марцинкевич В. А.

Минск 2022

1 КОД ПРОГРАММЫ

1.1 Пакет main

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "math/rand"
    "time"
)

func main() {
    var num string
    for {
        fmt.Printf("\nEnter number, which includes only 0 and
1(9 bit length): ")
        scanner := bufio.NewScanner(os.Stdin)
        scanner.Scan()
        num = scanner.Text()
        if len(num) <= 10 {
            break
        }
    }

    var bits [10]int
    index := 10 - len(num)

    for i := 0; i < index; i++ {
        bits[i] = 0
    }
    for _, s := range num {
        bits[index] = int(s - '0')
        index++
    }
    fmt.Printf("First num: %v\n", bits)

    var code [14]int
    var j int
    for i := 0; i <= 13; i++ {

        if i == 0 || i == 1 || i == 3 || i == 7 {
            code[i] = 0
        } else {
            code[i] = bits[j]
            j++
        }
    }
    fmt.Printf("Second number: %v\n", code)
```

```

var count int

if code[0] == 1 {
    count++
}
if code[2] == 1 {
    count++
}
if code[4] == 1 {
    count++
}
if code[6] == 1 {
    count++
}
if code[8] == 1 {
    count++
}
if code[12] == 1 {
    count++
}
count = count % 2
if count == 1 {
    code[0] = 1
}

count = 0
if code[1] == 1 {
    count++
}
if code[2] == 1 {
    count++
}
if code[6] == 1 {
    count++
}
if code[5] == 1 {
    count++
}
if code[9] == 1 {
    count++
}
if code[10] == 1 {
    count++
}
count = count % 2
if count == 1 {
    code[1] = 1
}

count = 0
if code[3] == 1 {

```

```

        count++
    }
    if code[4] == 1 {
        count++
    }
    if code[5] == 1 {
        count++
    }
    if code[6] == 1 {
        count++
    }
    if code[11] == 1 {
        count++
    }
    if code[12] == 1 {
        count++
    }
    count = count % 2
    if count == 1 {
        code[3] = 1
    }

    count = 0
    if code[7] == 1 {
        count++
    }
    if code[8] == 1 {
        count++
    }
    if code[9] == 1 {
        count++
    }
    if code[10] == 1 {
        count++
    }
    if code[11] == 1 {
        count++
    }
    if code[12] == 1 {
        count++
    }
    count = count % 2
    if count == 1 {
        code[7] = 1
    }

    fmt.Printf("Third number: %v\n", code)

    //decode
    err := code
    rand.Seed(time.Now().UnixNano())
    index = rand.Intn(13)

```

```

    if err[index] == 0{
        err[index] = 1
    }else{
        err[index] = 0
    }

    fmt.Printf("Error message: %v\n", err)

    j = 0
    var message [10]int
    if err[0] != code[0] || err[1] == code[1] || err[3] ==
code[3] || err[7] == code[7]{
        for i, n := range code{
            if i != 0 && i != 1 && i != 3 && i != 7{
                message[j] = n
                j++
            }
        }
        fmt.Printf("Correct message: %v\n", message)
        return
    }

    var count_pos int
    count = 0
    if err[0] == 1 {
        count++
    }
    if err[2] == 1 {
        count++
    }
    if err[4] == 1 {
        count++
    }
    if err[6] == 1 {
        count++
    }
    if err[8] == 1 {
        count++
    }
    if err[12] == 1 {
        count++
    }
    count = count % 2;
    if (count == 1 && err[0] == '0') || (count == 0 && err[0] ==
'1'){
        count_pos+=1
    }

    count = 0
    if err[1] == 1 {
        count++
    }

```

```

    }
    if err[2] == 1 {
        count++
    }
    if err[6] == 1 {
        count++
    }
    if err[5] == 1 {
        count++
    }
    if err[9] == 1 {
        count++
    }
    if err[10] == 1 {
        count++
    }
    count = count % 2
    if (count == 1 && err[1] == '0') || (count == 0 && err[1] ==
'1'){
        count_pos+=1
    }

    count = 0
    if err[3] == 1 {
        count++
    }
    if err[4] == 1 {
        count++
    }
    if err[5] == 1 {
        count++
    }
    if err[6] == 1 {
        count++
    }
    if err[11] == 1 {
        count++
    }
    if err[12] == 1 {
        count++
    }
    count = count % 2
    if (count == 1 && err[3] == '0') || (count == 0 && err[3] ==
'1'){
        count_pos+=1
    }

    count = 0
    if err[7] == 1 {
        count++
    }
    if err[8] == 1 {

```

```

        count++
    }
    if err[9] == 1 {
        count++
    }
    if err[10] == 1 {
        count++
    }
    if err[11] == 1 {
        count++
    }
    if err[12] == 1 {
        count++
    }
    count = count % 2
    if (count == 1 && err[7] == '0') || (count == 0 && err[7] ==
'1'){
        count_pos+=1
    }

    fmt.Printf("Error at pos: %v\n", count_pos)

    if err[count_pos] == 0 {
        err[count_pos] = 1
    }else{
        err[count_pos] = 0
    }

    j = 0
    for _, n := range err{
        if index != 0 && index != 1 && index != 3 && index != 7{
            message[j] = n
            j++
        }
    }
    fmt.Printf("Correct message: %v\n", message)
}

```

2 ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

1. Разработать программный модуль реализации защиты от ошибок кодом Хемминга .
2. Реализовать кодирование и декодирование информации.