

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«Самарский национальный исследовательский университет
имени академика С.П. Королёва» (Самарский университет)

Факультет информатики
Кафедра программных систем

Дисциплина
Теория информации

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

Шумозащитное кодирование

Вариант №7

Студент: Гижевская В.Д.

Группа: 6413-020302D

Преподаватель: Додонов М.В
Оценка:

Дата:

Самара 2021

ЗАДАНИЕ

1. Вычислить значение бита паритета к заданному информационному байту.
2. Построить код Хэмминга для информационного байта.
3. Укажите номер бита с ошибкой в коде Хемминга (использовать логические рассуждения и алгоритм Хэмминга).

ХОД РАБОТЫ

Задание 1

Информационный байт: 10100110

Бит паритета - 0

Задание 2

Пров. биты	Контролируемые биты											
1	1	3	5	7	9	11	13	15	17	19	21	...
2	2	3	6	7	10	11	14	15	18	19	22	...
4	4	5	6	7	12	13	14	15	20	21	22	...
8	8	9	10	11	12	13	14	15	24	25	26	...
16	16	17	18	19	20	21	22	23	24	25	26	...
32	32	33	34	35	36	37	38	39	40	41	42	...

Информационный байт: 10100110

1	2	3	4	5	6	7	8	9	10	11	12
0		1		0	1	0		0	1	1	0
0	0	1		0	1	0		0	1	1	0
0	0	1	1	0	1	0		0	1	1	0
0	0	1	1	0	1	0	0	0	1	1	0

1. Сумма чётная → 0
2. Сумма чётная → 0
3. Сумма нечётная → 1
4. Сумма чётная → 0

Задание 3

Информационный байт: 011100111011

1	2	3	4	5	6	7	8	9	10	11	12
0	1	1	1	0	0	1	1	1	0	1	1
0	1	1	1	0	0	1	1	1	0	1	1
0	1	1	1	0	0	1	1	1	0	1	1
0	1	1	1	0	0	1	1	1	0	1	1

1. Сумма чётная → ошибка в бите с чётным номером
2. Сумма чётная → ошибка в бите с номером 4, 8 или 12
3. Сумма нечётная → ошибка в бите с номером 4 или 12
4. Сумма чётная → ошибка в бите с номером 4

Правильный вариант: 011000111011

ЛИСТИНГ ПРОГРАММЫ

```
package com.company;

import java.util.Arrays;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Map;
import java.util.Set;

public class Main {

    static Map<Integer, Integer[]> indexes = new HashMap<>();
    static Map<Integer, Boolean> errors = new HashMap<>();

    public static void main(String[] args) {
        String infByte1 = "10100110";
        String infByte2 = "011100111011";

        putIn();

        System.out.println("\nЗадание 1");
        bitPariteta(infByte1);
        System.out.println("\nЗадание 2");
        codeHemminga(infByte1);
        System.out.println("\nЗадание 3");
        findIncorrect(infByte2);
    }

    public static void putIn() {
        indexes.put(1, new Integer[]{1, 3, 5, 7, 9, 11});
        indexes.put(2, new Integer[]{2, 3, 6, 7, 10, 11});
        indexes.put(4, new Integer[]{4, 5, 6, 7, 12});
        indexes.put(8, new Integer[]{8, 9, 10, 11, 12});

        errors.put(1, false);
        errors.put(2, false);
        errors.put(4, false);
        errors.put(8, false);
    }
}
```

```

public static void bitPariteta(String in) {
    boolean res = false;
    for (int i = 0; i < in.length(); i++) {
        if (in.charAt(i) == '1') {
            res = !res;
        }
    }
    System.out.println("Бит паритета = " + (res ? 1 : 0));
}

```

```

public static void codeHemminga(String in) {
    char[] res = new char[]{
        0, 0,
        in.charAt(0),
        0,
        in.charAt(1), in.charAt(2), in.charAt(3),
        0,
        in.charAt(4), in.charAt(5), in.charAt(6), in.charAt(7),
    };

    for (int i : indexes.keySet()) {
        boolean fig = false;
        for (int v : indexes.get(i)) {
            if (res[v - 1] == '1') {
                fig = !fig;
            }
        }
        res[i - 1] = fig ? '1' : '0';
    }

    System.out.println("Код ХЭММИНГА = " + Arrays.toString(res));
}

```

```

public static void findIncorrect(String in) {
    Set<Integer> errorIndex = new HashSet<>();
    char[] res = in.toCharArray();
    for (int i : indexes.keySet()) {
        boolean fig = false;
        for (int v : indexes.get(i)) {

```

```

        if (res[v - 1] == '1') {
            fig = !fig;
        }
    }
    char r = fig ? '1' : '0';
    errors.put(i, r != res[i - 1]);
}

for (int i : errors.keySet()) {
    if (errors.get(i)) {
        errorIndex.addAll(Arrays.asList(indexes.get(i)));
    }
}

for (int i : errors.keySet()) {
    if (!errors.get(i)) {
        errorIndex.removeAll(Arrays.asList(indexes.get(i)));
    }
}

System.out.println("Бит с ошибкой = " + errorIndex);
}
}

```