

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

Кафедра ИИТ

ОТЧЁТ

По лабораторной работе №1

«Избыточное кодирование данных в информационных системах. Код Хемминга»

Выполнил:
Студент группы ИИ-22
Гузареви́ч Д.А.
Проверил:
Хацкеви́ч А.С.

Цель работы: приобретение практических навыков кодирования/декодирования двоичных данных при использовании кода Хемминга.

Задание.

1. Закрепить теоретические знания по использованию методов помехоустойчивого кодирования для повышения надежности передачи и хранения в памяти компьютера двоичных данных.
2. Разработать приложение для кодирования/декодирования двоичной информации кодом Хемминга с минимальным кодовым расстоянием 3 или 4.
3. Результаты выполнения лабораторной работы оформить в виде отчета с листингом разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.
4. Ответить на контрольные вопросы

Ход работы

Вариант	M ¹	r
3	797	3

1. Составить код Хемминга (классический алгоритм) ($M+r$, M), допустить ошибку в одном из разрядов и отыскать её по алгоритму.
2. Составить код Хемминга (расширенный алгоритм) (первые 7 битов $M + 3$ проверочных, первые 7 битов M), допустить 2 или более ошибок в разрядах и отыскать их по алгоритму

Код программы:

```
#include <iostream>
#include <string>
#include <algorithm>
#include <vector>

std::vector<bool> BINARY(int num) {
    std::vector<bool> binary;

    while (num > 0) {
        binary.push_back(num % 2);
        num /= 2;
    }
    std::reverse(binary.begin(), binary.end());
    return binary;
}

std::string CONVERT_TO_STRING(std::vector<bool> binary) {
    std::string bin("");
    for (bool b : binary) {
        bin += std::to_string(b);
    }
    return bin;
}

int main() {
    setlocale(LC_ALL, "ru");

    printf("Введите число: ");
    int num = 0;
    std::cin >> num;
    std::vector<bool> bin = BINARY(num);
    std::reverse(bin.begin(), bin.end());

    for (int i = 0; i <= (int)sqrt(bin.size()); i++) {
        bin.insert(bin.begin() + pow(2, i) - 1, 0);
    }
}
```

```

}

printf("Исходная последовательность: %s\n", CONVERT_TO_STRING(bin).c_str());

// формирование дополнительных битов
int temp = 0;
for (int i = 0; i < bin.size(); i++) {
    if (bin[i] == 1) {
        temp ^= i + 1;
    }
}

std::vector<bool> additional_bits = BINARY(temp);
std::reverse(additional_bits.begin(), additional_bits.end());
printf("Дополнительные биты: %s\n", CONVERT_TO_STRING(additional_bits).c_str());

for (int i = 0, j = 0; i <= (int)sqrt(bin.size()); i++) {
    bin[pow(2, i) - 1] = additional_bits[j++];
}

printf("Отправленная последовательность: %s\n", CONVERT_TO_STRING(bin).c_str());

bin[2] = !bin[2];

printf("Последовательность с ошибкой: %s\n", CONVERT_TO_STRING(bin).c_str());

temp = 0;
for (int i = 0; i < bin.size(); i++) {
    if (bin[i] == 1) {
        temp ^= i + 1;
    }
}

printf("Позиция ошибочного бита: %d\n", temp - 1);

bin[temp - 1] = !bin[temp - 1];

printf("Исправленная последовательность: %s\n", CONVERT_TO_STRING(bin).c_str());
return 0;
}

```

Результат работы:

```

Введите число: 797
Исходная последовательность: 00100110100011
Дополнительные биты: 0001
Отправленная последовательность: 00100111100011
Последовательность с ошибкой: 00000111100011
Позиция ошибочного бита: 2
Исправленная последовательность: 00100111100011

```

Вывод: приобрёл практические навыки кодирования/декодирования двоичных данных при использовании кода Хемминга.