

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ”
ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Отчёт
по дисциплине
Современные методы защиты компьютерных систем
по лабораторной работе №1
«Избыточное кодирование данных в информационных системах.
Код Хемминга»

Выполнил:
Студент группы ИИ-22
Борейша О.С.
Проверил:
Хацкевич А. С.

Брест 2024

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

Задачи:

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

Код программы:

```
def hamming_code(m):
    r_positions = [1, 2, 4, 8]

    data_bits = list(map(int, m))
    code = [None] * 14

    j = 0
    for i in range(1, 15):
        if i in r_positions:
            code[i-1] = 0
        else:
            code[i-1] = data_bits[j]
            j += 1

    for r in r_positions:
        xor_sum = 0
        for i in range(1, 15):
            if i & r != 0:
                xor_sum ^= code[i - 1]
        code[r - 1] = xor_sum

    return code

def introduce_error(code, error_pos):
    code[error_pos - 1] ^= 1
    return code

def detect_and_correct_error(code):
    error_pos = 0
    for r in [1, 2, 4, 8]:
        xor_sum = 0
        for i in range(1, 15):
            if i & r != 0:
                xor_sum ^= code[i - 1]
        if xor_sum != 0:
            error_pos += r

    if error_pos != 0:
        print(f"Ошибка обнаружена на позиции {error_pos} ({bin(error_pos)[2:]})")
        code[error_pos - 1] ^= 1
    else:
        print("Ошибок не найдено.")
```

```
return code
```

```
M = list(bin(748)[2:])
print(f'Исходное сообщение: {M}')

hamming_encoded = hamming_code(M)
print("Закодированное сообщение Хэмминга:", hamming_encoded)

error_pos = 5
hamming_with_error = introduce_error(hamming_encoded.copy(), error_pos)
print("Сообщение с ошибкой:", hamming_with_error)

corrected_code = detect_and_correct_error(hamming_with_error.copy())
print("Исправленное сообщение:", corrected_code)
```

Результат работы:

```
Исходное сообщение: ['1', '0', '1', '1', '1', '0', '1', '1', '0', '0']
Закодированное сообщение Хэмминга: [0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0]
Сообщение с ошибкой: [0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0]
Ошибка обнаружена на позиции 5 (101).
Исправленное сообщение: [0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0]
```

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