

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
Сидоренко А.А.
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:

```
import math

def check_verf_b(M):
    return int(math.log(len(M), 2))

def insert_error(M, index):
    M[index - 1] ^= 1
    return M

def sum_p_bts(p_index, M):
    for i in p_index:
        sum_index = 0
        step = i + 1
        for j in range(i, len(M), 2 * step):
            sum_index += sum(M[j: j + step])
        M[i] = sum_index % 2

def encode(M):
    print("Кодирование...")
    print("Исходный код:", M)
    p_index = []
    for i in range(check_verf_b(M) + 1):
        index = 2 ** i - 1
        p_index.append(index)
        M.insert(index, 0)

    print("Код с нулевыми контрольными битами:", M)

    sum_p_bts(p_index, M)

    print("Код ХЭММИНГА:", M)

def decode(M):
    print("Декодирование...")
    p_index = []
    M_cp = M.copy()
    fls_ind = []

    for i in range(check_verf_b(M) + 1):
        index = 2 ** i - 1
        p_index.append(index)
```

```

        M[index] = 0

    sum_p_bts(p_index, M)

    for i in p_index:
        if M[i] != M_cp[i]:
            fls_ind.append(i + 1)
    return sum(fls_ind)

def main():
    M_bnr = 0b1011101010
    M = list(map(int, bin(M_bnr)[2:]))
    encode(M)
    error_ind = 2
    print("Код хэмминга с ошибкой в позиции:", error_ind, insert_error(M, error_ind))
    er_ind = decode(M)
    print("Ошибка в позиции:", er_ind)

if __name__ == "__main__":
    main()

```

Результат работы:

```

C:\Users\Xiaomi\AppData\Local\Programs\Python\Python312\python.exe C:\Users\Xiaomi\Desktop\prjs
Кодирование...
Исходный код: [1, 0, 1, 1, 1, 0, 1, 0, 1, 0]
Код с нулевыми контрольными битами: [0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0]
Код Хэмминга: [1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0]
Код хэмминга с ошибкой в позиции: 2 [1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0]
Декодирование...
Ошибка в позиции: 2

```

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