

Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

Кафедра интеллектуально-информационных технологий

Лабораторная работа №1

“Код Хемминга”

По дисциплине “Современные методы защиты компьютерных систем”

Выполнила:
студентка 4 курса
группы ИИ-22
Нестерчук Д. Н.
Проверил:
Хацкевич А.С.

Брест-2024

Цель работы: реализовать код Хемминга для своего варианта.

Вариант 11

Ход работы:

Задание. Составить код Хемминга (классический алгоритм) ($M+r$, M), допустить ошибку в одном из разрядов и отыскать её по алгоритму. $M = 590$, $r = 5$.

Код программы:

```
import math

r = 0

def encode_hamming_code(number):
    binary_number = bin(number)[2:]
    data_length = len(binary_number)

    global r
    r = 0
    while 2 ** r < (data_length + r + 1):
        r += 1

    code_length = data_length + r
    code = [0] * code_length
    XOR = 0

    j = 0
    for i in range(code_length):
        if math.log(i + 1, 2).is_integer():
            code[i] = 0
        else:
            code[i] = int(binary_number[j])
            j += 1
            if code[i] == 1:
                XOR ^= i + 1

    str_XOR = bin(XOR)[2:].zfill(r)

    for i in range(r):
        position = 2 ** i
        code[position - 1] = int(str_XOR[i])

    print("\nGenerated Hamming Code:")
```

```

print("".join(map(str, code)))
return code

def check_hamming_code(code):
    XOR = 0
    for i in range(len(code)):
        if code[i] == 1:
            XOR ^= i + 1

    if XOR != 0:
        print(f"Error detected at position: {XOR}")
        code[XOR - 1] ^= 1
        print("Corrected Hamming Code: ", end="")
        print("".join(map(str, code)))
    else:
        print("No errors detected.")

def decode_hamming_code(code):
    data_bits = "".join(str(code[i]) for i in range(len(code)) if not math.log(i + 1,
2).is_integer())
    return int(data_bits, 2)

# Пример использования
number = 25 # Исходное число
encoded_code = encode_hamming_code(number)
check_hamming_code(encoded_code)

# Допустим, ошибка была внесена в закодированное сообщение
encoded_code[4] ^= 1 # Симулируем ошибку
check_hamming_code(encoded_code)

decoded_number = decode_hamming_code(encoded_code)
print(f"Decoded number: {decoded_number}")

```

Результат работы программы:

```

Исходные данные: 1001001110
Закодированные данные: 0111001100111001
Данные с ошибкой: 0111101100111001
Ошибка обнаружена в бите 5, исправляем...
Исправленные данные: 0111001100111001

```

Вывод: Изучила и реализовала код Хемминга.