

Федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

Прикладная математика
Лабораторная работа №5

Выполнила: Калугина Марина

Группа: Р3302

г. Санкт-Петербург

2019 г.

Цель работы:

Получить практические навыки построения циклического кода по заданным характеристикам и проверка его свойства по обнаружению и исправлению ошибок.

Листинг программы:

```
#include<iostream>
#include<vector>
#include<string>
using namespace std;
// функция для того, чтобы из строки получить последовательность бит
vector<int> parseInt(string a) {
    vector<int> result(a.length(), 0);
    for (int i = 0; i < a.length(); i++)
        result[i] = a[i] == '0' ? 0 : 1;
    return result;
}

vector<int> add(vector<int> a, vector<int> b, int c=0) {
    vector<int> sum(a.size(), 0);
    while (b.size() != a.size())
        b.insert(b.begin(), 0);

    for (int i = sum.size() - 1; i >= 0; i--) {
        sum[i] = ((a[i] ^ b[i]) ^ c); // c is carry
        c = ((a[i] & b[i]) | (a[i] & c)) | (b[i] & c);
    }

    return sum;
}

// рассчитать код. Функция возвращает остаток от деления
vector<int> divide(vector<int> data, vector<int> crc) {
    vector<int> result = data;
    for (int i = 0; i < crc.size() - 1; i++)
        result.push_back(0);

    for (int i = 0; i <= result.size() - crc.size(); i) {
        // используем XOR для деления многочленов
        for (int j = 0; j < crc.size(); j++)
            result[i + j] = result[i + j] ^ crc[j];

        while (result[i] != 1) {
            i++;
            if (i == result.size())
                break;
        }
    }
}
```

```

    }

    return vector<int>(result.end() - crc.size() + 1, result.end());
}

int main() {
    string temp;

    cout << "Делитель: "; //1011
    cin >> temp;
    vector<int> crc = parseInt(temp);

    cout << "Информационные биты: "; //11010011101100
    cin >> temp;
    vector<int> data = parseInt(temp);

    vector<int> rem = divide(data, crc);
    cout << "Остаток: ";
    for (auto i : rem)
        cout << i;

    cout << "\nКорректная информация: ";
    vector<int> trans = data;
    trans.insert(trans.end(), rem.begin(), rem.end());
    for (auto i : trans)
        cout << i;

    cout << "\n\n\n=====\nПроверка\n=====\n";
    cout << "\nИнформационные биты, которые были получены алгоритмом: ";
    for (auto i : trans)
        cout << i;

    cout << "\nДелитель: ";
    for (auto i : crc)
        cout << i;

    rem = divide(trans, crc);

    bool valid = true;
    cout << "\nОстаток: ";
    for (auto i : rem){
        cout << i;
        if (i != 0)
            valid = false;
    }
}

```

```
cout << endl;
if (valid)
    cout << "\nВерно";
else
    cout << "\nНе верно";

cout << endl;
return 0;
}
```

Пример вывода программы:

Делитель: 1011

Информационные биты: 11010011101100

Остаток: 100

Корректная информация: 11010011101100100

=====

Проверка

=====

Информационные биты, которые были получены алгоритмом: 11010011101100100

Делитель: 1011

Остаток: 000

Верно

Вывод:

в ходе выполнения лабораторной работы я познакомилась с таким видом кодов, обнаруживающих одиночные ошибки, как циклический код и реализовала алгоритм расчета такого кода.