МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

**

**ОТЧЁТ**

**по лабораторной работе №7**

**«**Рекурсивные алгоритмы**»**

**по дисциплине: «***Программирование***»**

Выполнил:Проверил:

Студент гр. «АБс-324», «АВТФ» *Ассистент кафедры ЗИ*

*Тятюшкин М.О. Исаев Г. А.*

«\_\_» \_\_\_\_\_\_ 2024 г.«\_\_\_» \_\_\_\_\_\_ 2024 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (подпись)

Новосибирск 2024

**Цели и задачи работы:** изучение рекурсивного программирования, методов разработки эффективных алгоритмов.

**Задания:**

1. Напишите рекурсивную функцию, проверяющую правильность расстановки скобок в строке. При правильной расстановке выполняются условия: (а) количество открывающих и закрывающих скобок равно. (б) внутри любой пары открывающая – соответствующая закрывающая скобка, скобки расставлены правильно **Вариант 19.**
2. Дана закодированная строка, верните её декодированную версию. Правило кодирования: k[encoded\_string], где encoded\_string квадратные скобки повторяются ровно k раз. Обратите внимание, что k это гарантированно положительное целое число **Вариант 5**.

**Задание 1**

**C++**  
  
 #include <iostream>

#include <stack>

#include <string>

using namespace std;

bool Skobki(const string& str, int pos = 0, int opened = 0) {

    if (pos == str.size()) {

        return opened == 0;  // Проверяем, что все скобки были закрыты

    }

    if (str[pos] == '(') {

        return Skobki(str, pos + 1, opened + 1);

    } else if (str[pos] == ')') {

        if (opened > 0) {

            return Skobki(str, pos + 1, opened - 1);

        } else {

            return false;  // Найдена лишняя закрывающая скобка

        }

    } else {

        return Skobki(str, pos + 1, opened);  // Пропускаем символы, отличные от скобок

    }

}

int main() {

    setlocale (LC\_ALL, "rus");

    system ("chcp 65001");

    string input;

    cout << "Введите строку со скобками: ";

    cin >> input;

    if (Skobki(input)) {

        cout << "Скобки расставлены правильно." << endl;

    } else {

        cout << "Скобки расставлены неправильно." << endl;

    }

    return 0;

}

**Python:**

def Skobki(str, pos=0, opened=0):

    if pos == len(str):

        return opened == 0

    if str[pos] == '(':

        return Skobki(str, pos + 1, opened + 1)

    elif str[pos] == ')':

        if opened > 0:

            return Skobki(str, pos + 1, opened - 1)

        else:

            return False

    else:

        return Skobki(str, pos + 1, opened)

if \_\_name\_\_ == "\_\_main\_\_":

    input\_str = input("Введите строку со скобками: ")

    if Skobki(input\_str):

        print("Скобки расставлены правильно.")

    else:

        print("Скобки расставлены неправильно.")

**Задание 2**

**C++:**

#include <iostream>

#include <string>

using namespace std;

// Рекурсивная функция для декодирования закодированной строки

string decodeString(const string& s, size\_t& idx) {

    string result = "";

    int count = 0;

    while (idx < s.size()) {

        if (isdigit(s[idx])) {

            count = count \* 10 + s[idx] - '0';

        } else if (s[idx] == '[') {

            // Рекурсивный вызов для декодирования подстроки внутри квадратных скобок

            string decodedString = decodeString(s, ++idx);

            // Добавляем к результату количество раз декодированной подстроки

            for (int i = 0; i < count; ++i) {

                result += decodedString;

            }

            count = 0;

        } else if (s[idx] == ']') {

            return result; // Возвращаем результат, когда встречается ']'

        } else {

            result += s[idx]; // Добавляем символы к результату

        }

        ++idx;

    }

    return result; // Возвращаем окончательную декодированную строку

}

// Функция для декодирования закодированной строки посредством вызова вспомогательной функции

string decodeString(const string& s) {

    size\_t idx = 0;

    return decodeString(s, idx);

}

int main() {

    system ("chcp 65001");

    setlocale (LC\_ALL, "rus");

    string encodedString; // Закодированная строка для декодирования

    cout << "Encode string: " << encodedString;

    cin >> encodedString; // Ввод строки

    string decodedString = decodeString(encodedString); // Декодируем строку

    cout << "Decode string: " << decodedString << endl;

    return 0;

}

**Python:**

def decodeString(s: str) -> str:

    stack = []

    curr\_num = 0

    curr\_str = ''

    for char in s:

        if char.isdigit():

            curr\_num = curr\_num \* 10 + int(char)

        elif char == '[':

            stack.append((curr\_str, curr\_num))

            curr\_str, curr\_num = '', 0

        elif char == ']':

            prev\_str, num = stack.pop()

            curr\_str = prev\_str + curr\_str \* num

        else:

            curr\_str += char

    return curr\_str

# Примеры

print(decodeString("3[a]2[bc]"))

print(decodeString("3[a2[c]]"))

print(decodeString("2[abc]3[cd]ef"))

**Вывод:**

В ходе выполнения лабораторной работы №7 я познакомился с понятием рекурсии и научился реализовывать её на различных примерах.

**Ссылка на репозиторий:** [нажмите](https://github.com/astnq/laba7)