ОСАМИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Алгоритмы и структуры данных»

Тема: Рекурсия

Студент гр. 9382 _____ Русинов Д.А.

Преподаватель _____ Фирсов М.А.

Санкт-Петербург

2020

Цель работы.

Познакомиться с процессом рекурсии и построить синтаксический анализатор для понятия скобки.

Задание.

16. Построить синтаксический анализатор для понятия скобки. скобки::=A | B | (скобки скобки)

Алгоритм.

При вызове функции увеличивается счетчик глубины рекурсии функции.
При завершении соответственно уменьшается счетчик глубины рекурсии.

Проверяется переменная result, а также индекс и длина строки.
Если данные параметры не удовлетворяют требованиям, тогда происходит выход из функции.

Затем проверяется символ, который находится под текущим индексом
Если символ не является ни 'A', ни 'B', ни '(', тогда выражение не является скобками.

Если символ является 'A' или 'B', тогда необходимо проверить, является ли этот символ первым, и если является, необходимо проверить длину строки
Если символ первый и длина строки равна единице, тогда выражение – скобки.
Если длина строки не равна единице, тогда выражение не является скобками.

Если символ является '(', тогда необходимо проверить следующие два символа. Если они удовлетворяют выражению скобки, что проверяется повторным вызовом функции в цикле 2 раза, тогда проверяется символ после них, является ли он символом ')', и если все требования выполнены, тогда проверка окончена.

После проверки требований увеличивается проверяемый индекс строки, уменьшается глубина рекурсии и происходит выход из функции.

В конце проверяется, все ли символы были проверены. Если не все, тогда выражение не является скобками.

Название и описание функций.

- void print(const char* message) — функция, печатающая отступ, соответствующий глубине рекурсии, которую обозначает глобальная переменная currentDepth;
  - const char* message — сообщение, которое выведется на экран.

- void isBracket(std::string str, int* currentIndex, bool* result) — функция проверки выражения на соответствие понятию «скобки». Принимает аргументы:
  - std::string str — проверяемое выражение;
  - int* currentIndex — индекс текущего символа;
  - bool* result — переменная, содержащая результат работы функции;

- int main() – в этой функции считывается выражение и запускается основная логика, затем выводится результат работы программы на экран.

Тестирование

Входные данные	**Выходные данные**
AA	.Вызвана функция! .Первым символом был встречен 'A' | 'B', но он был не последним! Рекурсивная функция не дошла до конца строки! Данное выражение не является скобками!
((A(BA))(AB))	.Вызвана функция! .Встречен терминальный символ '(' ! ..Вызвана функция! ..Встречен терминальный символ '(' ! ...Вызвана функция! ...Был встречен символ 'A' | 'B'!

	...Вызвана функция! ...Встречен терминальный символ '(' ! ....Вызвана функция! ....Был встречен символ 'A' | 'B'! ....Вызвана функция! ....Был встречен символ 'A' | 'B'! ...Был встречен терминальный символ ')' ! ..Был встречен терминальный символ ')' ! ..Вызвана функция! ..Встречен терминальный символ '(' ! ...Вызвана функция! ...Был встречен символ 'A' | 'B'! ...Вызвана функция! ...Был встречен символ 'A' | 'B'! ..Был встречен терминальный символ ')' ! .Был встречен терминальный символ ')' ! Данное выражение является скобками!
(A(AB)A)	.Вызвана функция! .Встречен терминальный символ '(' ! ..Вызвана функция! ..Был встречен символ 'A' | 'B'! ..Вызвана функция! ..Встречен терминальный символ '(' ! ...Вызвана функция! ...Был встречен символ 'A' | 'B'! ...Вызвана функция! ...Был встречен символ 'A' | 'B'! ..Был встречен терминальный символ ')' ! .Не был встречен терминальный символ ')' ! Рекурсивная функция не дошла до конца строки! Данное выражение не является скобками!
()	.Вызвана функция! .Встречен терминальный символ '(' ! ..Вызвана функция! ..Не были обнаружены символы 'A', 'B', '(' !

	.Строка не является скобками! Рекурсивная функция не дошла до конца строки! Данное выражение не является скобками!
(((AB)B)((AB)A))	.Вызвана функция! .Встречен терминальный символ '(' ! ..Вызвана функция! ..Встречен терминальный символ '(' ! ...Вызвана функция! ...Встречен терминальный символ '(' ! ....Вызвана функция! ....Был встречен символ 'A' | 'B'! ....Вызвана функция! ....Был встречен символ 'A' | 'B'! ...Был встречен терминальный символ ')' ! ...Вызвана функция! ...Был встречен символ 'A' | 'B'! ..Был встречен терминальный символ ')' ! ..Вызвана функция! ..Встречен терминальный символ '(' ! ...Вызвана функция! ...Встречен терминальный символ '(' ! ....Вызвана функция! ....Был встречен символ 'A' | 'B'! ....Вызвана функция! ....Был встречен символ 'A' | 'B'! ...Был встречен терминальный символ ')' ! ...Вызвана функция! ...Был встречен символ 'A' | 'B'! ..Был встречен терминальный символ ')' ! .Был встречен терминальный символ ')' ! Данное выражение является скобками!
((A(BA))(AB))A	.Вызвана функция! .Встречен терминальный символ '(' ! ..Вызвана функция! ..Встречен терминальный символ '(' ! ...Вызвана функция! ...Был встречен символ 'A' | 'B'!

	...Вызвана функция! ...Встречен терминальный символ '(' ! ....Вызвана функция! ....Был встречен символ 'A' | 'B'! ....Вызвана функция! ....Был встречен символ 'A' | 'B'! ...Был встречен терминальный символ ')' ! ..Был встречен терминальный символ ')' ! ..Вызвана функция! ..Встречен терминальный символ '(' ! ...Вызвана функция! ...Был встречен символ 'A' | 'B'! ...Вызвана функция! ...Был встречен символ 'A' | 'B'! ..Был встречен терминальный символ ')' ! .Был встречен терминальный символ ')' ! Рекурсивная функция не дошла до конца строки! Данное выражение не является скобками!

Выводы.

Был изучен процесс рекурсии, и данное знание было применено для реализации синтаксического анализатора. Для этого была написана программа на языке программирования C++.

ПРИЛОЖЕНИЕ А.
ИСХОДНЫЙ КОД.

Текст файла main.cpp

```
#include <iostream>

/* Запуск программы из файла с перенаправлением потоков
 * ./main > output.txt < input.txt
 * Вывод в файл output.txt и ввод из файла input.txt
 *
 * Ввод из консоли, вывод в консоль
 * ./main
 */


int currentDepth = 0;

void print(const char* message) {
    std::string output = std::string();

    // Имитация питона. '.' * currentDepth
    // Для форматированного вывода, который отображает глубину рекурсии
    for (int i = 0; i < currentDepth; ++i) {
        output += '.';
    }
    output += message;
    std::cout << output << std::endl;
}

// Функция для проверки выражения.
void isBracket(std::string str, int* currentIndex, bool* result) {
    currentDepth += 1;
    print("Вызвана функция!");

    if (!(*result)) {
        print("Выражение не является скобками!");
        // Уменьшим глубину рекурсии
        currentDepth -= 1;
        return;
    }

    if (str.length() == *currentIndex) {
        print("Строка закончилась!");
        *result = false;
        // Уменьшим глубину рекурсии
        currentDepth -= 1;
        return;
    }

    // Сравним символ с '('
    if (str[*currentIndex] == '(') {
        print("Встречен терминальный символ '(' !");

        *currentIndex += 1;
        // Пойдем по строке дальше

        for (int i = 0; i < 2; ++i) {
            isBracket(str, currentIndex, result);
            if (!(*result)) {
```

```
                print("Строка не является скобками!");
                // Уменьшим глубину рекурсии
                currentDepth -= 1;
                return;
            }
        }

        if (str[*currentIndex] != ')') {
            print("Не был встречен терминальный символ ')' !");
            *result = false;
            // Уменьшим глубину рекурсии
            currentDepth -= 1;
            return;
        }

        print("Был встречен терминальный символ ')' !");

    } else if ( (str[*currentIndex] != 'A') && (str[*currentIndex] != 'B') )
{
        print("Не были обнаружены символы 'A', 'B', '(' !");
        *result = false;
        // Уменьшим глубину рекурсии
        currentDepth -= 1;
        return;

    } else {
        if ((*currentIndex == 0) && (str.length() != 1)) {
            print("Первым символом был встречен 'A' | 'B', но он был не по-
следним!");
            *result = false;
            // Уменьшим глубину рекурсии
            currentDepth -= 1;
            return;
        }
        print("Был встречен символ 'A' | 'B'!");
    }

    *currentIndex += 1;
    // Уменьшим глубину рекурсии
    currentDepth -= 1;
}


int main() {
    std::string str;
    std::cin >> str;
    int currentIndex = 0;
    bool result = true;
    isBracket(str, &currentIndex, &result);

    if ((result) && currentIndex == str.length()) {
        print("Данное выражение является скобками!");
    } else if (currentIndex != str.length()){
        print("Рекурсивная функция не дошла до конца строки!");
        print("Данное выражение не является скобками!");
    } else {
        print("Данное выражение не является скобками!");
    }

    return 0;
}
```