МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ФКТИ

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Алгоритмы и структуры данных»

Тема: РЕКУРСИЯ

Студент гр. 9303 _____ Федосихин Г. И.

Преподаватель _____ Филатов А. Ю.

Санкт-Петербург

2020

Цель работы.

Ознакомиться с основными понятиями и приёмами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++.

Постановка задачи.

Построить синтаксический анализатор для понятия скобки.

скобки::=квадратные | круглые

квадратные:: = [[квадратные] (круглые)] | B

круглые::=((круглые) [квадратные]) | A

Основные теоретические положения.

Рекурсивным называется объект, содержащий сам себя или определенный с помощью самого себя. Мощность рекурсии связана с тем, что она позволяет определить бесконечное множество объектов с помощью конечного высказывания. Бесконечные вычисления можно описать с помощью конечной рекурсивной программы. Рекурсивные алгоритмы лучше всего использовать, когда решаемая задача, вычисляемая функция или обрабатываемая структура данных определены с помощью рекурсии. Если процедура (или функция) P содержит явное обращение к самой себе, она называется прямо рекурсивной. Если P содержит обращение к процедуре (функции) Q, которая содержит (прямо или косвенно) обращение к P, то P называется косвенно рекурсивной. Многие известные функции могут быть определены рекурсивно. Например факториал, который присутствует практически во всех учебниках по программированию, а также наибольший общий делитель, числа Фибоначчи, степенная функция и др.

Выполнение работы.

В программе реализованы следующие функции:

1. Функция bool IsBracket(char* &ptr) принимает на вход указатель ptr типа char* на символьный массив str. Функция является входной точкой рекурсивного алгоритма программы. В зависимости от первого символа, функция вызывает одну из двух других функций, перечисленных ниже, которые обрабатывают набор символов и проверяют, соответствует ли входные данные заданным правилам построения скобок. Функция возвращает значения типа bool - результат обработки программы (является ли входная строка скобками или нет).

2. Функция bool IsRound(char*& ptr) принимает на вход указатель на символьный массив, из функции bool IsBracket(char* &ptr). Функция проверяет символ строки str на соответствие заранее определённому синтаксису скобок. Если символом круглой скобки является «A», то возвращается значение true. Если символом круглой скобки является «(», то функция проверяет, что последующие символы соответствуют конструкции «((A)[B])». В процессе проверки функция рекурсивно вызовет саму себя, а так же функцию bool IsSquare(char* &ptr). В случае, если все вложенные функции вернули true, и порядок скобок не нарушен, то функция возвращает true, иначе функция выводит на экран ошибку и возвращает false.

3. Функция bool IsSquare(char* &ptr) принимает на вход указатель на символьный массив, из функции bool IsBracket(char* &ptr). Функция проверяет символ строки str на соответствие заранее определённому синтаксису скобок. Если символом круглой скобки является «B», то возвращается значение true. Если символом круглой скобки является «[», то функция проверяет, что последующие символы соответствуют конструкции «[[B](A)]». В процессе проверки функция рекурсивно вызовет саму себя, а так же функцию bool IsRound(char*& ptr). В случае, если все вложенные функции вернули true, и порядок скобок не нарушен, то функция возвращает true, иначе функция выводит на экран ошибку и возвращает false.

6. Функция int main() принимает на вход строку символов, создает указатель на массив, вызывает функцию bool IsBracket(char* &ptr) и обрабатывает входные данные. После запуска программы пользователю предлагается ввести выражение, и программа напечатает на экран результат работы или сообщит об ошибке.

Выводы.

Была реализована программа, для синтаксического анализа выражения «скобки» согласно заранее определённым правилам. Программа включает в себя рекурсивные функции, благодаря которым осуществляется анализ.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <iostream>
bool IsSquare(char*& ptr);
bool IsRound(char*& ptr);

using namespace std;

bool IsSquare(char* &ptr) {
    if (*ptr == 'B') { ptr++; return true; }
    else if (*ptr++ == '[') {
        if (*ptr++ == '[') {
            if (IsSquare(ptr)) {
                if (*ptr++ == ']') {
                    if (*ptr++ == '(') {
                        if (IsRound(ptr)) {
                            if (*ptr++ == ')') {
                                if (*ptr++ == ']') {
                                    return true;
                                }
                                else
                                    cout << "Нет завершающей ]" <<
endl;
                            }
                            else
                                cout << "Нет завершающей )" <<
endl;
                        }
                        else
                            cout << "Круг ошибочен" << endl;
                    }
                    else
```

```
                        cout << "Нет (" << endl;
                    }
                    else
                        cout << "Нет ]" << endl;
                }
                else
                    cout << "Квадрат ошибочен" << endl;
            }
            else
                cout << "Нет второй начинающей [" << endl;
        }
        else {
            cout << "Не [ и не B" << endl;
            cout << *ptr << endl;
        }
        return false;
    }

    bool IsRound(char*& ptr) {
        if (*ptr == 'A') { ptr++; return true; }
        else if (*ptr++ == '(') {
            if (*ptr++ == '(') {
                if (IsRound(ptr)) {
                    if (*ptr++ == ')') {
                        if (*ptr++ == '[') {
                            if (IsSquare(ptr)) {
                                if (*ptr++ == ']') {
                                    if (*ptr++ == ')') {
                                        return true;
                                    }
                                    else
                                        cout << "Нет завершающей )" <<
    endl;
                                }
                                else
                                    cout << "Нет завершающей ]" <<
    endl;
                            }
                            else
                                cout << "Квадрат ошибочен" << endl;
                        }
                        else
                            cout << "Нет [" << endl;
                    }
                    else
                        cout << "Нет )" << endl;
                }
                else
                    cout << "Круг ошибочен" << endl;
            }
            else
                cout << "Нет второй начинающей (" << endl;
        }
```

```
    else
        cout << "Не ( и не A" << endl;
    return false;
}

bool IsBracket(char* &ptr){
    bool a = false;
    if (*ptr == 'B' || *ptr == '[') a = IsSquare(ptr);
    else if (*ptr == 'A' || *ptr == '(') a = IsRound(ptr);
    else
        cout << "Недопустимый начальный символ" << endl;
    if (a && *ptr != '\0') {
        cout << "Лишние символы" << endl;
        return false;
    }
    return true;
}

int main() {
    setlocale(LC_ALL, "rus");
    char str[256];;
    cout << "Введите предложение: " << endl;
    cin.getline(str, 256);
    char* ptr = str;
    bool a = false;

    if (*ptr != '\0') {
        if (IsBracket(ptr))
            cout << "ЭТО СКОБКИ" << endl;
        else
            cout << "ЭТО НЕ СКОБКИ" << endl;
    }
    else
        cout << "ПУСТО" << endl;
    return 0;
}
```

ПРИЛОЖЕНИЕ Б
ТЕСТИРОВАНИЕ

№	Входные данные	Выходные данные	Комментарий
1		ПУСТО	Тест обработки пустой строки
2	Z	Недопустимый	Тест обработки

		начальный символ	некорректных символов.
3	AA	Лишние символы	Тест обработки лишних символов.
4	[	Нет второй начинающей [	Тест обработки неполной конструкции скобок.
5	[[B	Нет]	Тест обработки неполной конструкции скобок.
6	[[]]	Ошибка вложенной скобки	Тест обработки неполной конструкции скобок.
7	A	ЭТО СКОБКИ	Тест обработки примитивных конструкций скобок.
8	((A)[B])	ЭТО СКОБКИ	
9	[[[[B](A)]](((A)[B]))]	ЭТО СКОБКИ	Тест обработки сложных конструкций скобок.
10	[[[[[[B](A)]](((A)[B]))]](A)]	ЭТО СКОБКИ	