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

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Алгоритмы и структуры данных»

Тема: Рекурсия

Студент гр. 9384 _____ Прашутинский К.И.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Разработать алгоритм рекурсивного прохода по строке с целью выяснения корректности строки.

Задание.

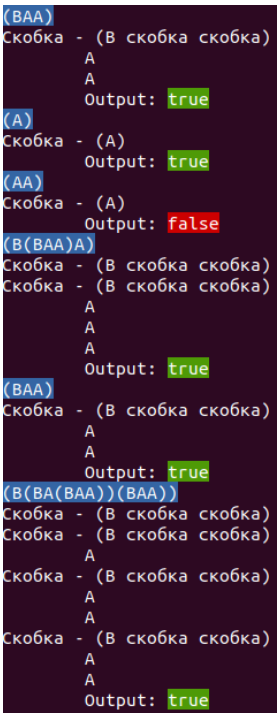
Построить синтаксический анализатор для понятия скобки. скобки::=A | (B скобки скобки).

Экспериментальные результаты.

Ввод:

```
(BAA)
(A)
(AA)
(B(BAA)A)
(BAA)
(B(BA(BAA))(BAA))
```

Вывод:



Обработка результатов эксперимента.

В ходе выполнения работы была создана функция Bracket(std::string &str, char c, int &i) рекурсивного прохода строки, возвращающая значение типа bool, в котором, в зависимости от строки выбирался один из вариантов работы программы.

Выводы.

Получен практический опыт по рекурсии и считыванию строки из файла.

ПРОТОКОЛ

```
#include <iostream>
#include <cstring>
#include <fstream>

bool Bracket(std::string &str, char c, int &i) {
	bool b = false;
	int count = 0;
	if (c == '(') {
		std::cout << "Скобка - (";
		i++;
		c = str[i];
		/*if (c == 'A') {
			std::cout << "A)\n";
			i++;
			c = str[i];
			if (c != ')') {
				i++;
				c = str[i];
				b = Bracket(str, c, i);
			}
			else {
				b = true;
			}
		}*/
		//else {
			if (c == 'B') {
				std::cout << "B скобка скобка)\n";
				i++;
```

```
                        c = str[i];
                        b = Bracket(str, c, i);

                        if (b == true) {
                                i++;
                                c = str[i];
                                b = Bracket(str, c, i);
                                if (b) {
                                        i++;
                                        c = str[i];
                                        if (c == ')') {
                                                b = true;
                                        }
                                        else b = false;

                                }
                        }

                }
        //}

    }
    else {
        if (c == 'A') {
                std::cout << "\tA\n";
                b = true;
        }
    }
    return b;
}
```

```
int main()
{

	int i = 0, count = 0;
	std::string str;
	std::ifstream stream("examples.txt");
	if(stream.is_open())
	while(std::getline(stream, str)){
		std::cout << "\x1b[0;44m" << str << "\x1b[0m\n";
		while(str[count] != '\0') count++;
		char c = str[i];
		if(Bracket(str, c, i) && (i + 1 == count || (count == 1 && i == 0)))
			std::cout << "\tOutput: \x1b[0;42mtrue\x1b[0m\n";
		else
			std::cout << "\tOutput: \x1b[0;41mfalse\x1b[0m\n";
		i = 0;
		count = 0;
	}
	else std::cout << "The input file is not open!\n";
	return 0;
}
//"(A(BAA))"
//"(B(BA(BAA))A)"
//"(B(BAA)(BAA))"
```