

МИНОБРНАУКИ 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

Студентка гр. 9383

Лапина А.А.

Преподаватель

Попова Е.В.

Санкт-Петербург

2020

Цель работы.

Ознакомиться с основными понятиями и приемами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций.

Задание.

Вариант 16.

Построить синтаксический анализатор для понятия скобки.

скобки::=A | B | (скобки скобки)

Теория.

Рекурсия - способ организации вычислительного процесса, при котором процедура или функция в ходе выполнения составляющих ее операторов обращается сама к себе.

Ход работы:

1. Проанализировать данные задачи, выделив рекурсивные действия с данными.
2. Составить схему выполнения программы.
3. Разработать программу с использованием рекурсии.
4. Протестировать программу

Выполнение работы:

В данной задаче нам дается выражение, а нам необходимо определить является ли оно скобкой — если да, то выводим «True», в противном случае «False». Чтобы лучше разораться с написанием программы, создадим схему: если подается строка A или B, то сразу возвращаем значение 1, если подается строка, в которой первый символ = «(», а последний = «)», то удаляем эти (внешние) скобки и уже выполняем следующие проверки (так как есть ситуации - «скобки в скобках») - делим на 3 вида:

* 1 вариант, когда первый символ **не** «(» , а последний — любой (пример входных данных: (A B) или (A (A B))), так как внешние скобки удалили, то этот вариант будет искать выражения : A B или A (A B), именно поэтому прописываем условие, что первый символ не является «(», а последний может быть скобкой, а может и нет, алгоритм в обоих случаях одинаков, поэтому 2 типа объединили с целью сокращения кода и простоты понимания алгоритма) ;

* 2 вариант, когда первый символ = «(», а последний **не** «)» - пример: ((A B) A)

* 3 вариант, когда 1 символ = «(» и последний = «)» - пример: ((A B) (A B)).

Таким образом, мы постоянно дробим строчку на маленькие части, каждую из которых рекурсивно проверяем и перемножаем значения, так как, если 1 часть не удовлетворяет условию, то получается и все выражение не является скобкой.

Пример работы программы:

1) Входные данные:

A

Выходные данные:

True

```
a2@a2-Lenovo-ideapad-330-15ARR:~/документы/alg_str$ ./pr
Введите строку
A
True
a2@a2-Lenovo-ideapad-330-15ARR:~/документы/alg_str$
```

2) Входные данные:

((A B) B)

Выходные данные:

True

```
a2@a2-Lenovo-ideapad-330-15ARR:~/документы/alg_str$ ./pr
Введите строку
((A B) B)
True
```

3) Входные данные:

(((A B) B) A)

Выходные данные:

True

```
a2@a2-Lenovo-ideapad-330-15ARR:~/документы/alg_str$ ./pr
Введите строку
(((A B) B) A)
True
```

4) Входные данные:

((A B) (A B))

Выходные данные:

True

```
a2@a2-Lenovo-ideapad-330-15ARR:~/документы/alg_str$ ./pr
Введите строку
((A B) (A B))
True
a2@a2-Lenovo-ideapad-330-15ARR:~/Документы/alg_str$
```

5) **Входные данные:**

(A)

Выходные данные:

False

```
a2@a2-Lenovo-ideapad-330-15ARR:~/документы/alg_str$ ./pr
Введите строку
(A)
False
a2@a2-Lenovo-ideapad-330-15ARR:~/Документы/alg_str$
```

6) **Входные данные:**

(A B (

Выходные данные:

False

```
a2@a2-Lenovo-ideapad-330-15ARR:~/документы/alg_str$ ./pr
Введите строку
(A B (
False
```

Разработанный программный код см. в приложении А.

Вывод.

Было произведено знакомство с основными понятиями и приемами рекурсивного программирования, были получены навыки программирования рекурсивных процедур и функций на языке программирования С.

Приложение А

Исходный код программы

Название файла: a.cpp

```
#include <iostream>
#include <cctype>
#include <cstring>

using namespace std;

int Rec(char a[20]){
    char a1[20];
    char a2[20];
    int i = 0;
    int j = 0;
    int ki = 0;
    int ko = 0;
    if (strcmp(a, "A") == 0 || strcmp(a, "B") == 0)
        return 1;
    //считаем длину
    int len = 0;
    while (a[len]!='\0')
        len++;
    if (a[0]=='(' && a[len-1] == ')')&& len>3){
        //удаляем скобки
        for (i = 0; i<len-1; i++){
            a[i] = a[i+1];
        }
        a[len-2] = '\0';
        i=0;
        //рассматриваем случай 1 вида:(A B) или (A (A B))
        if ((a[0]!='(' && a[len-3]!=')') || (a[0]!='(' && a[len-
3]==')')){
            while (a[i]!=' '){
                a1[i] = a[i];
                i++;
            }
            a1[i] = '\0';
            i++;
            while (a[i]!='\0'){
                a2[j] = a[i];
                j++;
                i++;
            }
            a2[j] = '\0';
            return Rec(a1)*Rec(a2);
        }
        //рассмотрим случай 2 вида: ((A B) A)
        if (a[0]=='(' && a[len-3]!=')'){
            for (i=0; i<len-4; i++){
                a1[i]=a[i];
            }
            a1[i] = '\0';
            a2[0] = a[i+1];
            a2[1] = '\0';
            return Rec(a1)*Rec(a2);
        }
    }
```

```

        //рассмотрим случай 3 вида: ((A B) (A B))
        if (a[0]=='(' && a[len-3]==')'){
            i = 0;
            while(a[i]!=''){
                if (a[i]=='(')
                    ki++;
                i++;
            }
            i = 0;
            while (ko!=ki){
                a1[i] = a[i];
                if (a[i]==')')
                    ko++;
                i++;
            }
            a1[i] = '\\0';
            i++;
            j = 0;
            while (a[i]!='\\0'){
                a2[j] = a[i];
                j++;
                i++;
            }
            a2[j] = '\\0';
            return Rec(a1)*Rec(a2);
        }
    }
    return 0;
}

int main(){
    cout << "Введите строку" << "\\n";
    char a[20];
    int i = 0;
    a[0] = getchar();
    while (a[i]!='\\n'){
        i++;
        a[i] = getchar();
    }
    a[i] = '\\0';
    int rezult;
    rezult = Rec(a);
    if (rezult==1)
        cout << "True" << "\\n";
    else
        cout << "False" << "\\n";
    return 0;
}

```