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. 7382 _____ Гиззатов А.С.

Преподаватель _____ Фирсов М.А.

Санкт-Петербург

2018

Задание.

Варинт №14.

Построить синтаксический анализатор для понятия *скобки.*

скобки::=A | (B *скобки скобки*)

Описание алгоритма:

Были написаны 2 функции:1)удаление всех табуляций и пробелов(void no_tabs(char* str){),2)рекурсивная функция(br_check(char* str,int cur,int k)),3)Main функция int main(){.

1)Функция no_tabs получает на вход строку и удаляет все табуляции и пробелы. Функция проходит по всем символам и проверяет их на табуляцию и пробел, если символ является пробелом то функция запоминает адрес следующего символа и затирает символ с табуляцией. После склеивает остаток 1 строки и записанный адрес.

2)Функция br_check получает на вход:1)строку,2)номер символа строки, который нужно обработать,3) глубина рекурсии. Функция ищет подстроку в строке,которая удовлетворяет условию и равна (BAA), и заменяет его на A. После переходит на 2 символа назад и повторяет все предыдущие действия пока не останется строка равная (BAA), тогда функция возвращает 1, если же функция не может выполнить данное задание тогда она возвращает 0.

3)Функция main считывает текст из входного потока,обрабатывает его, выводит результат в выходной поток.

Тестирование.

Были написаны 10 тестов для данной программы, а также скрипт для тестирования и компиляции программы.

Результаты тестирования на рис.1. и рис.2

В 10 тесте можно увидеть скобки вида (B(B(B(B(BAA)A)A)A)(B(B(B(BAA)A)A)A)). После пошагового выполнения программы этот текст имел вид 1) (B(B(B(BAA)A)A)(B(B(BAA)A)A)) после 1 выполнения рекурсивной функции, 2)(B(B(BAA)A)(B(BAA)A)) после 2 выполнения функции, 3) (B(BAA)(BAA)) после 3 выполнения функции, 4) (B(BAA)(BAA))

после 4 выполнения функции, 5) (BAA) после 5 выполнения функции, 6) функция вернула ноль так как данная комбинация символов подходит под одно из условий функции.

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

struct dic{
        char *word;
        char *translation;
};


int main(){
        struct dic dict;
        int i,k=0;
        dict = (struct dic)calloc(100,sizeof(struct dic));
        fgets("%s %s", dict.word, dict.translation);
        while(dict.word = "ugabuga"){
                k++;
                scanf("%s %s",dict.word, dict.translation);
        }
        char *str = calloc(100,sizeof(char));
        fgets(str,100,stdin);
        str[strlen(str)] = "\0";
        char *txt = strtok(str," ");
        while(txt != NULL){
                for(i=0;i<k;i++){
                        if(strcmp(dict.word,txt) == 0){
                                pritnf("%s",dict.translation);
                        }
                }
                if(strcmp(dict.word,txt) != 0){
                        printf("<unknown>");
                }
                txt=strtok(str," ");
        }



        return 0;
}
```

Рисунок 1

```
вывод 6 теста:

Текст после удаления всех пробелов и табуляций:(B(BAA)(BAA))
Вывод изменения текста после выполнения каждого шага рекурсии:
Глубина рекурсии:0
Текст после выполнения алгоритма:(BA(BAA))
        Глубина рекурсии:1
        Текст после выполнения алгоритма:(BAA)
Результат-текст является скобками
____________________
                тест#7:
 (      B       A       A)
вывод 7 теста:

Текст после удаления всех пробелов и табуляций:(BAA)
Вывод изменения текста после выполнения каждого шага рекурсии:
Результат-текст является скобками
____________________
                тест#8:
(B(B(B(B(BAA)A)A)A)(BAA))
вывод 8 теста:

Текст после удаления всех пробелов и табуляций:(B(B(B(B(BAA)A)A)A)(BAA))
Вывод изменения текста после выполнения каждого шага рекурсии:
Глубина рекурсии:0
Текст после выполнения алгоритма:(B(B(B(BAA)A)A)(BAA))
        Глубина рекурсии:1
        Текст после выполнения алгоритма:(B(B(BAA)A)(BAA))
                Глубина рекурсии:2
                Текст после выполнения алгоритма:(B(BAA)(BAA))
                        Глубина рекурсии:3
                        Текст после выполнения алгоритма:(BA(BAA))
                                Глубина рекурсии:4
                                Текст после выполнения алгоритма:(BAA)
Результат-текст является скобками
```

рис.2

```
вывод 9 теста:

Текст после удаления всех пробелов и табуляций:(B(B(B(B(B(B(B(B((B()B)B)BB(B()
Вывод изменения текста после выполнения каждого шага рекурсии:
Результат-текст не является скобками
____________________
                тест#10:
(B(B(B(B(BAA)A)A)A)(B(B(B(BAA)A)A)A))
вывод 10 теста:

Текст после удаления всех пробелов и табуляций:(B(B(B(B(BAA)A)A)A)(B(B(B(BAA)A)A)A))
Вывод изменения текста после выполнения каждого шага рекурсии:
Глубина рекурсии:0
Текст после выполнения алгоритма:(B(B(B(BAA)A)A)(B(B(B(BAA)A)A)A))
        Глубина рекурсии:1
        Текст после выполнения алгоритма:(B(B(BAA)A)(B(B(B(BAA)A)A)A))
                Глубина рекурсии:2
                Текст после выполнения алгоритма:(B(BAA)(B(B(B(BAA)A)A)A))
                        Глубина рекурсии:3
                        Текст после выполнения алгоритма:(BA(B(B(B(BAA)A)A)A))
                                Глубина рекурсии:4
                                Текст после выполнения алгоритма:(BA(B(B(BAA)A)A))
                                        Глубина рекурсии:5
                                        Текст после выполнения алгоритма:(BA(B(BAA)A))
                                                Глубина рекурсии:6
                                                Текст после выполнения алгоритма:(BA(BAA))
                                                        Глубина рекурсии:7
                                                        Текст после выполнения алгоритма:(BAA)
Результат-текст является скобками
```

рис.3

Выводы.

В результате работы были усвоены методы использования рекурсии, а также написана программа с использование метода рекурсии.

Исходный код

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#define N 100

int br_check(char* str,int cur,int k){
	//cur=0;
	char* ptr;
	if(strcmp(str,"(BAA)") == 0){
		return 1;
	}
	if(cur>strlen(str)){
		return 0;
	}
	if(strcmp(str,"A") == 0){
		return 1;
	}
	if(str[cur] == '(' && str[cur+1] == 'B' && str[cur+2] == 'A' && str[cur+3]
== 'A' && str[cur+4] == ')'){
		str[cur] = 'A';
		//deleting unnecessary symbols
		ptr = str+cur+5;
	// and replacing them with single 'A'
		str[cur+1]='\0';
		strcat(str,ptr);
		for(int i=0;i<k;i++)
			printf("\t");
		printf("Глубина рекурсии:%d\n",k);
		for(int i=0;i<k;i++)
				printf("\t");
```

```
            printf("Текст после выполнения алгоритма:%s\n",str);
            br_check(str,cur-2,k+1);
            //call recursion with
      }
      else{
            br_check(str,cur+1,k);
      }
}

void no_tabs(char* str){
      char* cur_ptr;
      for(int i=0;i<strlen(str);i++){
            if(isspace(str[i])){
            //deleting all tabulations and spaces
                  cur_ptr = str+i+1;
                  str[i]= '\0';
                  strcat(str,cur_ptr);
                  i--;
            }
      }
}




int main(){
        char *brackets = (char*)calloc(N,sizeof(char));
        fgets(brackets,N,stdin);
            //reading the brackets
        brackets[strlen(brackets)-1]='\0';
        no_tabs(brackets);   //deleting all the spaces and tabulations
      printf("Текст после удаления всех пробелов и табуляций:%s\n",brackets);
      printf("Вывод изменения текста после выполнения каждого шага
рекурсии:\n");


        if(br_check(brackets,0,0)){
                printf("Результат-текст является скобками\n");
        }
        else{
                printf("Результат-текст не является скобками\n");
```

```
        }
    free(brackets);
        return 0;
}
```