

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. 7383

Русецкий И.И.

Преподаватель

Размочаева Н.В.

Санкт-Петербург

2018

Содержание

1. Цель работы	3
2. Реализация задачи	4
3. Тестирование	6
3.1 Процесс тестирования.....	6
3.2 Результаты тестирования.....	6
4. Вывод	7
Приложение А. Тестовые случаи	8
Приложение Б. Исходный код	9

1. ЦЕЛЬ РАБОТЫ

Цель работы: познакомиться с основными понятиями и приемами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++.

Формулировка задачи: Вариант 17. Функция Φ преобразования текста определяется следующим образом (аргументы функции – это текст, т.е. последовательность символов и область в памяти): $\Phi(\gamma)\beta$, если

$$\Phi(\alpha) = \begin{cases} \overline{\alpha} = \beta/\gamma \text{ и тест } \beta \text{ не содержит символов вхождений символа } \gamma, \\ \alpha, \text{ если в } \alpha \text{ нет вхождений символа } \gamma. \end{cases}$$

2. РЕАЛИЗАЦИЯ ЗАДАЧИ

Для решения поставленной задачи было принято создать функцию `strtok_(char *str, char *mem)`, которая принимает аргумент строку (последовательность символов) и область в памяти размером 4 байта (в нее можно записать адрес функции `strtok_1`), она использует рекурсию, которая происходит пока есть символы разделители в строке, а затем выводит в обратном порядке лексемы.

Функция `char *strtok_1(char *str, char *delim, char **mem)` реализована для разделения строки на лексемы.

Функция `func(char str, char *delim)` реализована для возвращения определенного значения в зависимости от сравнения символа `str` с другими.

Функция `void keyboard_input(char **mem)` реализована для ввода строки с клавиатуры и применение над ней функции `strtok_`.

Функция `void file_input(char **mem)` реализована для вывода действия работы функции `strtok_` над строкой из файла.

Функция `void Interface()` реализована для использования программы, т.е. меню.

Работа алгоритма:

Функция `strtok_` выполняет поиск лексем в строке `str` и выводит лексемы в обратном порядке. Последовательность вызовов этой функции разбивают

строку `str` на лексемы, которые представляют собой последовательности символов, разделенных символами-разделителями.

На первый вызов, функция принимает строку `str` в качестве аргумента, чей первый символ используется в качестве начальной точки для поиска лексем. В последующие вызовы, функция ожидает нулевого указателя и использует позицию сразу после окончания последней лексемы как новое местонахождение для сканирования.

Для определения начала лексемы функция сначала определяет символы, не содержащиеся в строке(символы-разделители). А затем посимвольно проверяет остальную часть строки до первого символа-разделителя, который сигнализирует конец лексемы.

Этот конечный маркер автоматически заменяется нулевым символом, и лексема возвращается функцией. После этого, следующие вызовы функции `strtok_` начинаются с этого нулевого символа. В самом конце выводит все лексемы в обратном порядке.

3. ТЕСТИРОВАНИЕ

3.1 ПРОЦЕСС ТЕСТИРОВАНИЯ

Программа собрана в операционной системе Ubuntu 18.04.1 LTS bionic компилятором g++ (Ubuntu 7.3.0-16 ubuntu3) 7.3.0. В других ОС и компиляторах тестирование не проводилось.

3.2 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Тестовые случаи представлены в Приложении А.

Во время тестирования была обнаружена ошибка: программа выводила лишние символы после выполнения операции. В программу была добавлена переменная, которая зависит от другой функции, чтобы исключить данную ошибку.

4. ВЫВОД

В ходе выполнения данной работы были изучены основные принципы рекурсивного программирования. Были созданы функции для решения поставленной задачи рекурсивным методом.

ПРИЛОЖЕНИЕ А. ТЕСТОВЫЕ СЛУЧАИ

Строка	Строка(символы-разделители)	Вывод
но/ти/ра/бу	/	буратино
а/к/с/и/с/о/с	/	сосиска
5/4/3/2/1	/	12345
ук/па/-/ек/ов/чел	/	человек-паук

ПРИЛОЖЕНИЕ Б. ИСХОДНЫЙ КОД

```
#include<iostream>

#include<cstring>

#include<fstream>

#define MAX_LENGTH 1000

using namespace std;

int func(char str,char *delim);
char *strtok_1(char *str,char *delim,char **mem);
void strtok_(char *str,char **mem);
void keyboard_input(char **mem);
void file_input(char **mem);
void Interface();

int main(){
Interface();

int number=0;

char *mem=0;

cin>>number;

while(number){
```

```

switch(number){
case 1:
    keyboard_input(&mem);
    cout<<endl;
    Interface();
    cin>>number;
    break;
case 2:
    file_input(&mem);
    cout<<endl;
    Interface();
    cin>>number;
    break;
default:
    int a=3;
    if(number==a)return 0;
    cout<<"Неверно введены данные!"<<endl;
    Interface();
    cin>>number;
}
}

return 0;

```

```
}
```

```
int func(char str,char *delim){
```

```
int i=0;
```

```
for(;i<strlen(delim);i++)
```

```
{
```

```
if(str==delim[i])
```

```
    return 1;
```

```
if(str=='\0')
```

```
    return 2;
```

```
}
```

```
return 0;
```

```
}
```

```
char *strtok_1(char* str,char *delim,char **mem){
```

```
char *pch=*mem;
```

```
int i=0;
```

```
if(str!=NULL){
```

```
    *mem=str;
```

```
    pch=*mem;
```

```

}

if(*mem==NULL)

return NULL;

while(func(*pch,delim)==0)

pch++;

i=func(*pch,delim);

*pch='\0';

pch++;

swap(pch,*mem);

if(i==2)

*mem= NULL;

return pch;

}

```

```

void strtok_(char *str,char **mem){

char *pch=str;

if((pch=strtok_1(str,(char*)" /",mem))!=NULL){

strtok_(NULL,mem);

cout<<pch;

}

}

```

```
void keyboard_input(char **mem){  
    char str[MAX_LENGTH];  
    cout<<"Введите строку:"<<endl;  
    cin>>str;  
    strtok_(str,mem);  
}
```

```
void file_input(char **mem){  
    string filename;  
    char str[MAX_LENGTH];  
    cout<<"Имя файла:"<<endl;  
    cin>>filename;  
    ifstream file;  
    file.open(filename.c_str());  
    if(!file){  
        cout<<"Нет такого файла!"<<endl;  
    }  
    else{  
        cout<<"Есть такой файл!"<<endl;  
        file.getline(str,MAX_LENGTH);  
    }  
}
```

```
cout<<str<<endl;  
strtok_(str,mem);  
file.close();  
}
```

```
void Interface(){  
    cout<<"Выберите действие: "<<endl;  
    cout<<"1:Ввод с клавиатуры"<<endl;  
    cout<<"2:Вывод из файла"<<endl;  
    cout<<"3:Выход"<<endl;  
}
```