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

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 Ф преобразования текста определяется следующим образом (аргумент функции – это текст, т.е. последовательность символов): Ф(γ)β, если

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

2. РЕАЛИЗАЦИЯ ЗАДАЧИ

Для решения поставленной задачи было принято создать функцию strtok_(char *str),которая принимает аргумент строку(последовательность символов),она использует рекурсию, которая происходит пока есть символы разделители в строке, а затем выводит в обратном порядке лексемы.

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

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

Функция void Interface() реализована как 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 7.3.0-16ubuntu3) 7.3.0. В других ОС и компиляторах тестирование не проводилось.

3.2 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Тестовые случаи представлены в Приложении А.

Во время тестирования не было обнаружено ошибок.

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

5. ПРИЛОЖЕНИЕ Б: ИСХОДНЫЙ КОД

```
#include<iostream>
#include<cstring>
#include<fstream>
#define MAX_LENGTH 1000

using namespace std;

void strtok_(char *str);
void keyboard_input();
void file_input();
void Interface();

int main(){
Interface();
int number=0;
cin>>number;
 while(number){
  switch(number){
  case 1:
   keyboard_input();
   cout<<endl;
```

```
   Interface();
   cin>>number;
   break;
  case 2:
   file_input();
   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;
}

void strtok_(char *str){
char *istr=str;
```

```
if((str=strtok(istr,"/"))!=NULL){
 strtok_(NULL);
 cout<<str;
 }
}

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

void file_input(){
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;
```

```
 return;
 }
 else{
 cout<<"Есть такой файл!"<<endl;
 file.getline(str,MAX_LENGTH);
 }
cout<<str<<endl;
strtok_(str);
file.close();
}

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