МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ФАКУЛЬТЕТ МАТЕМАТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ І ТЕОРІЇ СИСТЕМ УПРАВЛІННЯ

**ЗВІТ**

**З НАВЧАЛЬНОЇ (КОМП’ЮТЕРНОЇ) ПРАКТИКИ**

**Виконав:**

студент 1 курсу денної форми навчання спеціальності 122 Комп’ютерні науки ОП: «Сучасні інформаційні технології та програмування»

Савченко Анатолій Андрійович

**Керівник практики:**

Римар П. В., старший викладач

**Оцінка:**

\_\_\_\_\_\_\_\_\_\_ (балів)

\_\_\_\_\_\_\_\_\_\_ (ECTS)

\_\_\_\_\_\_\_\_\_\_ (національна)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_\_р.

Вінниця – 2019

Завдання № 5. Робота з символами та словами

Дана послідовність, що містить від 1 до 37 слів, у кожному з яких є від 1 до 7 рядкових латинських букв, між сусідніми словами – кома, за останнім словом крапка. Надрукувати: а) слова, що мають максимальну довжину; 16 б) усі слова, які зустрічаються в послідовності по одному разу; в) усі однакові слова, що зустрічаються не менш ніж 2 рази, указавши для кожного з них число його входжень у послідовність; г) усі слова за абеткою. Послідовність символів читати з файлу

Зчитуємо рядок з файлу і розділяємо на слова,

далі передаємо у відповідні функції.

А) Знаходимо довжину максимального слова і виводимо усі слова з такою довжиною

Б) Перевіряємо кожне слово на предмет повторів

В) Перевіряємо кожне слово і рахуємо кількість його повторів що ще не були підраховані

Г) Сортуємо бульбашкою

#include <iostream>

#include <fstream>

#include <cstring>

#include <algorithm>

using namespace std;

void A(char\* words[8],int N){

cout<<"a) "<<N<<" ";

string str;

int max;

for(int i=0;i<N;i++){

str=words[i];

if(str.length()>max){

max=str.length();

}

}

for(int i=0;i<N;i++){

str=words[i];

if(str.length()==max){

cout<<str<<"\t";

}

}

cout<<endl;

}

void B(char\* words[8],int N){

cout<<"b) ";

string str;

bool fl;

for(int i=0;i<N;i++){

str=words[i];

fl=false;

for(int j=0;j<N;j++){

if(str==string(words[j]) && i!=j)

{

fl=true;

break;

}

}

if(!fl)

cout<<str<<" ";

}

cout<<endl;

}

void C(char\* words[8],int N){

cout<<"c) ";

string str;

int k;

for(int i=0;i<N;i++){

str=words[i];

k=1;

for(int j=0;j<N;j++){

if(str==string(words[j]) && i<j)

{

k++;

}else if(str==string(words[j]) && i>j){

k=1;

break;

}

}

if(k>=2)

cout<<endl<<str<<" "<<k;

}

cout<<endl;

}

void D(char\* words[8],int N){

cout<<"d) ";

char\* temp;

for (int i = 0; i < N - 1; i++) {

for (int j = 0; j < N - i - 1; j++) {

if (string(words[j]) > string(words[j+1])) {

temp = words[j];

words[j] = words[j + 1];

words[j + 1] = temp;

}

}

}

for(int i=0;i<N;i++){

cout<<words[i]<<" ";

}

cout<<endl;

}

int main(){

ifstream in("data.txt");

string put;

getline(in,put);

char\* input=new char[put.length() + 1];

strcpy(input, put.c\_str());

in.close();

char\* words[8];

char\* point;

string str;

point=strtok(input," ,.");

int i=0;

while(point!=NULL){

words[i]=point;

point=strtok(NULL," ,.");

i++;

}

int N=i;

A(words,N);

B(words,N);

C(words,N);

D(words,N);

delete [] input;

delete [] point;

return 0;

}

