**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №3**

**по дисциплине «Программирование»**

Тема: "Указатели и матрицы"

|  |  |  |
| --- | --- | --- |
| Студент гр. 0324 |  | Гордиенко Т.Е. |
| Преподаватель |  | Глущенко А.Г |

Санкт-Петербург

2020

**Цель работы:** научиться работать со строками

**Постановка задачи**:

Необходимо написать программу, которая реализует поставленную задачу:

1)    С клавиатуры или с файла (\*) (пользователь сам может выбрать способ ввода) вводится последовательность, содержащая от 1 до 50 слов, в каждом из которых от 1 до 10 строчных латинских букв и цифр. Между соседними словами произвольное количество пробелов. За последним символом стоит точка.

2)    Необходимо отредактировать входной текст:

·        удалить лишние пробелы;

·        удалить лишние знаки препинания (под «лишними» подразумевается несколько подряд идущих знаков (обратите внимание, что «…» - корректное использование знака) в тексте);

·        исправить регистр букв, если это требуется (пример некорректного использования регистра букв: пРиМЕр);

3) Выполнить задание по варианту:

|  |  |
| --- | --- |
| 5 | Вывести на экран только те слова последовательности, в которых первая буква слова встречается в этом слове еще раз. |

4) Выполнить задание по варианту:

|  |  |
| --- | --- |
| 5 | Вывести на экран ту же последовательность, переместив все цифры, содержащиеся в словах, в конец соответствующих слов. |

5)  Необходимо найти все подстроки, которую введёт пользователь в имеющейся строке. Реализуйте два алгоритма: первый алгоритма – Линейный поиск, а второй алгоритм согласно вашему номеру в списке. Четные номера должны реализовать алгоритм КНМ, а нечетные – Бойера-Мура. (\*)

**Выполнение работы:**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

void read\_file(char s[]) {

char stroka[1000];

string path = "Text.txt";

ifstream file;

file.open(path);

if (!file.is\_open()) {

std::cout << "ошибка открытия файла";

return;

}

else {

file.getline(s, 1000, '\0');

}

}

int main() {

setlocale(0, "");

char stroka[1000];

std::cout << "Способ чтения: " << '\n';

std::cout << "1) В консоли" << '\n';

std::cout << "2) Из файла" << '\n';

int var;

cin >> var;

switch (var) {

case 1:

cin.getline(stroka, 1000, '.');

break;

case 2:

read\_file(stroka);

break;

default:

std::cout << "error" << '\n';

return 1;

break;

}

string s = stroka;

//Редактирование строки

bool slovo = 0;

if ((s[0] >= 'A' && s[0] <= 'z') || (s[0] >= 'А' && s[0] <= 'я')) {

slovo = 1;

}

for (int i = 1; i != s.length() - 1; i++) {

if ((s[i] < 'A' || s[i] > 'Z') && (s[i] < 'a' || s[i] > 'z') && (s[i] < 'А' || s[i] > 'я')) {

slovo = 0;

if (s[i] == '.' && s[i-1] == '.' && s[i+1] == '.') {

i++;

}

else if (s[i] == s[i - 1]){

s.erase(i, 1);

i--;

}

}

else {

if (s[i] >= 'A' && s[i] <= 'Z' && slovo == 1) {

s[i] += 'Z' - 'A' + 7;

}

else if (s[i] >= 'А' && s[i] <= 'Я' && slovo == 1) {

s[i] += 'Я' - 'А' + 1;

}

slovo = 1;

}

}

std::cout << s << '\n' << '\n';

//Вывод слов, в которых первая буква встречается ещё раз

int pos = 0;

char temp = '\0';

slovo = 0;

for (int i = 0; i <= s.length() - 1; i++) {

if (slovo == 0 && ((s[i] >= 'A' && s[i] <= 'Z') || (s[i] >= 'a' && s[i] <= 'z') || (s[i] >= 'А' && s[i] <= 'я'))) {

slovo = 1;

temp = s[i];

pos = i;

continue;

}

if (s[i] == temp && slovo == 1) {

while (((s[pos] >= 'A' && s[pos] <= 'Z') || (s[pos] >= 'a' && s[pos] <= 'z') || (s[pos] >= 'А' && s[pos] <= 'я')) && pos <= s.length() - 1) {

cout << s[pos];

pos++;

}

slovo = 0;

i = pos;

cout << '\n';

}

else if (s[i] == ' ') {

slovo = 0;

}

}

cout << '\n';

//Перемещение чисел из слов в их конец

slovo = 0;

int temp\_i;

string temp\_s;

for (int i = 0; i <= s.length() - 1; i++) {

if (((s[i] >= 'A' && s[i] <= 'Z') || (s[i] >= 'a' && s[i] <= 'z') || (s[i] >= 'А' && s[i] <= 'я'))) {

slovo = 1;

}

else if (slovo == 1 && s[i] >= '0' && s[i] <= '9') {

temp\_s.append(1, s[i]);

s.erase(i, 1);

i--;

}

else {

slovo = 0;

s.insert(i, temp\_s);

temp\_s.clear();

}

}

cout << s << '\n';

//Поиск подстрок

cout << "Введите подстроку" << '\n';

getline(cin, temp\_s);

getline(cin, temp\_s);

slovo = 1;

pos = 0;

for (int i = 0; i != s.length() - temp\_s.length(); i++) {

for (int j = i; j != i + temp\_s.length(); j++) {

if (s[j] == temp\_s[pos]) {

pos++;

}

else {

pos = 0;

slovo = 0;

break;

}

}

if (slovo == 1) {

cout << i << " ";

pos = 0;

}

else {

slovo = 1;

}

}

return 0;

}