

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра «Информационные системы»

ОТЧЕТ
по практической работе №4
по дисциплине «Программирование»
Тема: текстовые строки как массивы символов

Студент гр. 0323

Балашевич К.Д.

Преподаватель

Глущенко А.Г.

Санкт-Петербург

2020

Цель работы.

Изучение способов обработки текстовых данных. Получение практических навыков работы с файлами. Изучение различных алгоритмов поиска подстроки в строке.

Основные теоретические положения.

Для работы с текстовыми строками в языке c++ используется унаследованный из языка си тип данных char, хранящий ascii-код символа и массивы char, для которых в заголовочном файле cstring (string.h) определены функции для упрощенной работы:

- strlen
- strcpy
- strcat
- strcmp
- и другие...

Фактическое завершение данных в строке указывается путём записи непосредственно после памяти нуль-терминатора ('\0').

Присутствует также возможность работать с объектами класса string.

Для записи и чтения данных из строк предусмотрена возможность работать с потоками данных, работа с которыми осуществляется при помощи заголовочного файла iostream. При этом в данном файле предопределены упрощенные способы работы с консолью cout, cin др.

Для чтения данных из потоков можно использовать класс ifstream.

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) Вывести на экран слова последовательности в алфавитном порядке.

3) Вывести на экран количество символов в каждом слове исходной последовательности.

5) Необходимо найти все подстроки, которую введёт пользователь в имеющейся строке. Реализуйте алгоритм линейного поиска, а второй алгоритм согласно вашему номеру в списке.

Сделать вывод по проделанной работе.

Выполнение работы.

Для решения поставленной задачи была написана программа на языке C++. Итоговый код программы представлен в приложении А. При написании программы было решено не использовать класс `string` в связи с меньшей эффективностью.

Было проведено тестирование программы с использованием различных ЭВМ и компиляторов. Результаты тестирования представлены в приложении Б. При этом были получены результаты, соответствующие расчётным.

Выводы.

Работа с си-строкой принципиально не отличается от работы с любыми другими массивами.

ПРИЛОЖЕНИЕ А

ПОЛНЫЙ КОД ПРОГРАММЫ

```
#include <iostream>
#include <cstring>
#include <cstdlib>
#include <fstream>
#include <windows.h>
#include <ctime>
#define Stringlength 1000

using namespace std;

int readTextFromTerminal (char String[]);
int readTextFromFile (char String[]);
int mainMenu (char String[], char stringOriginal[]);
int isCharASpace (char Char);
int isCharASign (char Char);
int isCharACapital (char Char);
int isCharADot (char Char);
void stringRedactor (char String[]);
void StringCut (char String[], int position, int number);
int alphabeticalSort (char String[], char sortedString[51][11]);
void numberSymbols (char String[]);
void linearSearch (char String[]);

int main ()
{
    //ВНИМАНИЕ!!! В СЛУЧАЕ ПРОБЛЕМ С ОТОБРАЖЕНИЕМ МЕНЮ
    РАСКОММЕНТИРУЙТЕ ДРУГУЮ СТРОЧКУ SETLOCALE!!!
    setlocale(LC_ALL, "ru_RU.utf8");
    //setlocale (0, "");
    //setlocale (LC_ALL, "Russian");
    char String[Stringlength];
    char stringOriginal[Stringlength];
    int exit=1;
    system("cls");
    cout << "Откуда прочитать строку:" << endl << "1) из терминала."
<< endl <<
        "2) из файла." << endl << endl << "Введите желаемый
вариант: ";
    int choise;
    cin >> choise;
```

```

cin.get();
switch (choise)
{
    case 1:
    {
        readTextFromTerminal(String);
        break;
    }
    case 2:
    {
        int result=readTextFromFile(String);
        if (!result)
        {
            system ("cls");
            cout << "Ошибка чтения файла!";
            Sleep (1000);
            return 1;
        }
        break;
    }
}
strcpy(stringOriginal, String);
while (exit) exit=mainMenu(String, stringOriginal);
return 0;
}

int mainMenu (char String[], char stringOriginal[])
{
    int choise;
    cout << "Текущее состояние строки:" << endl << endl;
    cout << String;
    cout << endl;
    cout << endl << "Главное меню работы со строкой:" << endl << "1)
Атоматичес"
        "ки отредактировать строку." << endl << "2) Восстановить
исходную с"
        "троку." << endl << "3) Вывести на экран слова
последовательности в"
        " алфавитном порядке." << endl << "4) Вывести на экран
количество с"
        "имволов в каждом слове исходной последовательности." <<
endl <<
        "5) Найти подстроки с помощью линейного поиска." << endl
<< "0) Вых"

```

```

        "од из программы." << endl << "Введите желаемый вариант:
";
    cin >> choise;
    switch (choise)
    {
        case 0:
            return 0;
        case 1:
            {
                stringRedactor(String);
                return 1;
            }
        case 2:
            {
                strcpy(String, stringOriginal);
                return 1;
            }
        case 3:
            {
                system ("cls");
                char sortedString[51][11];
                int sortedStringElements=alphabeticalSort(String,
sortedString);
                cout << "Отсортированные в алфавитном порядке слова:" <<
endl;
                for (int i=0; i<sortedStringElements+1; i++) cout <<
sortedString[i] << " ";
                cout << endl << endl;
                return 1;
            }
        case 4:
            {
                system ("cls");
                cout << "Число символов в каждом слове:" << endl;
                numberSymbols(String);
                cout << endl << endl;
                return 1;
            }
        case 5:
            {
                system ("cls");
                linearSearch(String);
                cout << endl << endl;
                return 1;
            }
    }

```

```

        default:
        {
            system("cls");
            cout << "Некорректный ввод!";
            Sleep(1000);
            return 1;
        }
    }
}

int readTextFromTerminal (char String[])
{
    cin.getline (String, 1000);
    cin.sync();
    return 1;
}

int readTextFromFile (char String[])
{
    char Name[Stringlength];
    cout << endl << "Введите имя (или путь) файла: ";
    cin.clear();
    cin.sync();
    cin.getline (Name, Stringlength);
    ifstream File;
    File.open (Name);
    if (!File) return 0;
    File.getline (String, Stringlength);
    File.close();
    return 1;
}

int isCharASpace (char Char)
{
    if (((int)Char)-32) return 0;
    return 1;
}

int isCharASign (char Char)
{
    if
        (((((int)Char>32)&&((int)Char<48))&&((int)Char-
46))||(((int)Char>57)&&((int)Char<65))||(((int)Char>90)&&((int)Char<97))|
|(((int)Char>122)&&((int)Char<128))) return 1;
    return 0;
}

```

```

    }

int isCharACapital (char Char)
{
    if (((int)Char>64)&&((int)Char<91)) return 1;
    return 0;
}

int isCharADot (char Char)
{
    if (((int)Char)-46) return 0;
    return 1;
}

void stringRedactor (char String[])
{
    int isFirst=1;
    int Length=strlen(String);
    for (int i=0; i < Length; i++)
    {
        if (!(isFirst))
            if (isCharACapital(String[i]))
            {
                String[i]=(int)String[i]+32;
            };
        isFirst=0;
        if (isCharASpace(String[i]))
        {
            int number=0;
            while (isCharASpace(String[i+1+number])) number++;
            if (number) StringCut (String, i+1, number);
            isFirst=1;
        }
        if (isCharASign(String[i]))
        {
            int number=0;
            while
(isCharASign(String[i+1+number])||isCharADot(String[i+1+number]))
number++;

            if (number) StringCut (String, i+1, number);
            isFirst=1;
        }
        if (isCharADot(String[i]))
        {

```



```

        int ellipsis=0;
        for (int j=i+1; j<i+3; j++)
        {
            if (isCharADot(String[j])) ellipsis++;
        }
        if (ellipsis==2)
        {
            i=i+1;
        }
        else
        {
            int number=0;
            while
(isCharASign(String[i+1+number])||isCharADot(String[i+1+number]))
number++;
            if (number) StringCut (String, i+1, number);
        }
        isFirst=1;
    }
}

void StringCut (char String[], int position, int number)
{
    int i;
    for      (i=position;          i<strlen(String);          i++)
String[i]=String[i+number];
}

int alphabeticalSort (char String[], char sortedString[51][11])
{
    char stringMod[Stringlength];
    strcpy(stringMod, String);
    int Length=strlen(stringMod);
    for (int i=0; i < Length; i++)
    {
        if
((isCharASign(stringMod[i]))||(isCharADot(stringMod[i]))) stringMod[i]='
';
    }
    for (int i=0; i < Length; i++)
        if (isCharASpace(stringMod[i]))
        {
            int number=0;

```

```

        while (isCharASpace(stringMod[i+1+number])) number++;
        if (number) StringCut (stringMod, i+1, number);
    };
    if (isCharASpace(stringMod[0])) StringCut (stringMod, 0, 1);
    int sortedStringElements=0, k=0;
    for (int i=0; i<strlen(stringMod); i++)
    {
        if (isCharASpace(stringMod[i]))
        {
            sortedString[sortedStringElements][k]='\0';
            sortedStringElements++;
            k=0;
        }
        else
        {
            sortedString[sortedStringElements][k]=stringMod[i];
            k++;
        }
    }
    sortedString[sortedStringElements][k]='\0';
    int sorted=0;
    int currentMaxUnsorted=sortedStringElements-1;
    while (!sorted)
    {
        sorted=1;
        for (int i=0; i<currentMaxUnsorted; i++)
        {
            if (strcmp(sortedString[i], sortedString[i+1])>0)
            {
                char buffer[11];
                strcpy(buffer, sortedString[i]);
                strcpy(sortedString[i], sortedString[i+1]);
                strcpy(sortedString[i+1], buffer);
                sorted=0;
            }
        }
        currentMaxUnsorted--;
    }
    return sortedStringElements;
}

void numberSymbols (char String[])
{
    char sortedString[51][11];

```

```

        int sortedStringElements=alphabeticalSort (String,
sortedString);
        cout << endl;
        for (int i=0; i<sortedStringElements+1;i++) cout <<
strlen(sortedString[i]) << "\t" << sortedString[i] << endl;
        cout << endl;
    }

void linearSearch (char String[])
{
    char searchString[Stringlength];
    cin.get();
    cout << endl << "Введите подстроку для поиска: ";
    cin.getline(searchString, Stringlength);
    int result=-1;
    cout << endl << "Искомая подстрока содержится на следующих
диапозонах символов: ";
    for (int i=0; String[i] ; i++)
        for (int j=0; ; j++)
        {
            if (!searchString[j])
            {
                result=i;
                cout << i+1 << "-" << i+j << " ";
                break;
            }
            if(String[i+j]!=searchString[j]) break;
        };
    if (result<0) cout << endl << "Таких подстрок нет!";
}

```

ПРИЛОЖЕНИЕ Б

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Тестирование программы осуществлялось на компьютере с процессором x64 под управлением ОС Windows 10 Pro (версия 2004, 64-бит) с использованием следующих компиляторов:

- g++ (GCC) 9.3.0
- Оптимизирующий компилятор Microsoft (R) C/C++ версии 19.16.27043 для x86

Результаты тестирования программы представлены в табл. 1.

Таблица 1 – Результаты тестирования программы

Исходная строка:		
One advanced dIverted doMestic sex rePEated bringing you old/&]]. Possible procured her trifling laughter thoughts property she met way. CompaNIons shy had solicitude favourable own. Which could saw ./?*]./ !@.@! !@... .. guest man now heard but.		
Действие:	Ожидаемый результат:	Фактический результат:
Ввод строки с помощью терминала	One advanced dIverted doMestic sex rePEated bringing you old/&]]. Possible procured her trifling laughter thoughts property she met way. CompaNIons shy had solicitude favourable own. Which could saw ./?*]./ !@.@! !@... .. guest man now heard but.	One advanced dIverted doMestic sex rePEated bringing you old/&]]. Possible procured her trifling laughter thoughts property she met way. CompaNIons shy had solicitude favourable own. Which could saw ./?*]./ !@.@! !@... .. guest man now heard but.
Ввод строки с помощью файла	One advanced dIverted doMestic sex rePEated bringing you old/&]]. Possible procured her trifling laughter thoughts property she met way. CompaNIons shy had solicitude favourable own. Which could saw ./?*]./ !@.@! !@... .. guest man now heard but.	One advanced dIverted doMestic sex rePEated bringing you old/&]]. Possible procured her trifling laughter thoughts property she met way. CompaNIons shy had solicitude favourable own. Which could saw ./?*]./ !@.@! !@... .. guest man now heard but.
Обработка строки	One advanced diverted domestic sex repeated bringing you old/ Possible procured her trifling laughter thoughts property she met	One advanced diverted domestic sex repeated bringing you old/ Possible procured her trifling laughter thoughts property

	way. Companions shy had solicitude favourable own. Which could saw . ! ! . guest man now heard but.	she met way. Companions shy had solicitude favourable own. Which could saw . ! ! . guest man now heard but.
Сортировка в алфавитном порядке	CompaNIons One Possible Which advanced bringing but could dIverted doMestic favourable guest had heard her laughter man met now old own procured property rePEated saw sex she shy solicitude thoughts trifling way you	CompaNIons One Possible Which advanced bringing but could dIverted doMestic favourable guest had heard her laughter man met now old own procured property rePEated saw sex she shy solicitude thoughts trifling way you
Вывод количества символов	10 CompaNIons 3 One 8 Possible 5 Which 8 advanced 8 bringing 3 but 5 could 8 dIverted 8 doMestic 10 favourable 5 guest 3 had 5 heard 3 her 8 laughter 3 man 3 met 3 now 3 old 3 own 8 procured 8 property 8 rePEated 3 saw 3 sex 3 she 3 shy	10 CompaNIons 3 One 8 Possible 5 Which 8 advanced 8 bringing 3 but 5 could 8 dIverted 8 doMestic 10 favourable 5 guest 3 had 5 heard 3 her 8 laughter 3 man 3 met 3 now 3 old 3 own 8 procured 8 property 8 rePEated 3 saw 3 sex 3 she 3 shy

	10 solicitude 8 thoughts 8 trifling 3 way 3 you	10 solicitude 8 thoughts 8 trifling 3 way 3 you
Поиск подстроки “ea”	287-288	287-288