

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра экономической математики, информатики и статистики (ЭМИС)

ОБРАБОТКА ТЕКСТОВЫХ СТРОК

Отчёт по лабораторной работе по дисциплине “Введение в
программирование”

Студент гр. 543-1



/А.Е. Мухамеджан

“ ____ ” ____ 2023

Доцент кафедры ЭМИС

____ Е.А. Шельмина
оценка

“ ____ ” ____ 2023 г.

Томск 2023

Лабораторная работа №7

“Обработка текстовых строк”

Цель работы: получить навыки обработки текстовых строк с использованием языка C/C++.

Представление символьной информации в ЭВМ: символьная информация хранится и обрабатывается в компьютере в виде цифрового кода. Для разных видов компьютеров и операционных систем используются различные наборы символов. Необходимый набор символов, предусмотренный в том или ином компьютере, обычно включает:

1. управляющие символы;
2. цифры;
3. буквы алфавита;
4. специальные знаки;
5. знаки операций;

Для представления символьной информации в компьютере чаще всего используется алфавит, состоящий из 256 символов. Каждый символ такого алфавита можно закодировать 8 битами.

Среди наборов символов наибольшее распространение получила таблица кодировки ASCII. В данной таблице стандартом являются только первые 128 символов. Сначала размещаются управляющие символы. Далее идут буквы латинского алфавита, цифры, знаки препинания, скобки и некоторые другие символы. Остальные 128 кодов таблицы символов ASCII используются для размещения символов национальных алфавитов, символов псевдографики и научных символов.

Для представления символьных переменных в языке Си используется тип `char`. Значением переменной типа `char` является 8-битный код, соответствующий тому или иному символу.

Библиотека обработки символов: библиотека обработки символов (`<ctype.h>`) содержит функции, выполняющие проверки и операции с

символьными данными. Каждая функция получает в качестве аргумента целое число, которое должно быть значением unsigned char или EOF. Список функций представлен на рисунке 1.

Прототип	Описание функции
int isdigit(int c)	проверка, является ли символ <i>c</i> десятичной цифрой
int islower(int c)	проверка, является ли символ <i>c</i> латинской буквой нижнего регистра ('a' – 'z')
int isupper(int c)	проверка, является ли <i>c</i> латинской буквой верхнего регистра ('A' – 'Z')
int isalpha(int c)	проверка, является ли <i>c</i> латинской буквой (isalpha(c) = islower(c) isupper(c))
int isalnum(int c)	буква либо цифра (isalnum(c) = isalpha(c) isdigit(c))
int tolower(int c)	перевод латинского символа <i>c</i> на нижний регистр
int toupper(int c)	перевод латинского символа <i>c</i> на верхний регистр
int isspace(int c)	проверка, является ли <i>c</i> пробелом (' '), сменой страницы ('\f'), новой строкой ('\n'), возвратом каретки ('\r'), табуляцией ('\t'), вертикальной табуляцией ('\v')
int iscntrl(int c)	проверка, является ли <i>c</i> управляющим символом
int isprint(int c)	проверка, является ли <i>c</i> печатаемым символом, включая пробел
int isgraph(int c)	проверка, является ли <i>c</i> печатаемым символом, кроме пробела
int ispunct(int c)	проверка, является ли <i>c</i> печатаемым символом, кроме пробела, буквы или цифры

Рисунок 1 – иллюстрация списка функции в библиотеке обработки символов

Строки в языке Си: в языке Си нет такого отдельно типа данных, как строка. Но можно определить строку двумя различными способами.

Строка как массив символов: во-первых, строку можно определить как массив символов, который заканчивается нулевым символом “\0”. Нуль символ используется для того, чтобы отмечать конец строки. В таблице символов ASCII данный символ имеет номер 0. При определении строки как массива символов ей присваивается имя и указывается максимальное количество символов, которое может содержаться в ней с учётом нулевого символа.

При объявлении строка может быть сразу же инициализирована, то есть она может быть присвоена массиву символов: char s[] = “moon”.

Подчеркнём, что наличие нулевого символа означает, что количество элементов массива символов должно быть по крайней мере на один больше, чем максимальное количество символов, планируемых для размещения в памяти.

Строка как указатель на первый символ: строку можно также определить и другим способом – с использованием указателя на символ. Если объявить `char *ps`, то этим самым задаётся переменная `ps`, которая может содержать адрес первого символа строки. В этом случае не происходит резервирование памяти для хранения символов, как в случае с массивом символов, и сама переменная `ps` не инициализируется конкретным значением.

Функции обработки строк: ниже приведены основные функции операций над строками (библиотека `<string.h>`) (рисунок 2).

Функции преобразования строк: ниже приводятся функции преобразования строк (библиотека `<stdlib.h>`) (рисунок 3).

Ход работы:

Задания для самостоятельной работы:

Задание 1. Требуется выделить в строке-предложении `s` все слова, разделенные символами разделителями «`_.,:;\n\t!?`» и обработать выделенные слова в соответствии с вариантом задания.

Для выполнения задания 1 (приложение А) использовался вариант 12, представленный в лабораторной работе №7 и интегрированная среда разработки Microsoft Visual Studio.

Задание 2. Даны два слова (две переменные). Сколько раз во втором слове встречается первая буква первого слова.

Для выполнения задания 2 (приложение Б) использовался вариант 12, представленный в лабораторной работе №7 и интегрированная среда разработки Microsoft Visual Studio.

Задание 3. Дан текст. Найти максимальное из имеющихся в нём чисел.

Для выполнения задания 3 (приложение В), представленного в лабораторной работе №7 использовалась интегрированная среда разработки Microsoft Visual Studio.

Вывод: в результате выполнения лабораторной работы получены навыки обработки текстовых строк с использованием языка C/C++..

Прототип	Описание функции
<code>int strlen(const char *s)</code>	возвращает длину строки <i>s</i>
<code>char *strcpy(char *s, const char *t)</code>	копирует строку <i>t</i> в строку <i>s</i> , включая '\0'; возвращает <i>s</i>
<code>char *strncpy(char *s, const char *t, size_t n)</code>	копирует не более <i>n</i> символов строки <i>t</i> в <i>s</i> ; возвращает <i>s</i> . Дополняет результат символами '\0', если символов в <i>t</i> меньше, чем <i>n</i>
<code>char *strcat(char *s, const char *t)</code>	приписывает <i>t</i> к <i>s</i> ; возвращает <i>s</i>
<code>char *strncat(char *s, const char *t, size_t n)</code>	приписывает не более <i>n</i> символов <i>t</i> к <i>s</i> , возвращает <i>s</i>
<code>int strcmp(const char *s, const char *t)</code>	сравнивает <i>s</i> и <i>t</i> ; возвращает <0, если <i>s</i> < <i>t</i> , 0, если <i>s</i> == <i>t</i> , и >0, если <i>s</i> > <i>t</i>
<code>int strncmp(const char *s, const char *t, size_t n)</code>	аналогична функции <code>strcmp()</code> , только сравниваются не более <i>n</i> первых символов в строках <i>s</i> и <i>t</i>
<code>char *strchr(const char *s, int c)</code>	возвращает указатель на первое вхождение символа <i>c</i> в строку <i>s</i> , либо NULL, если такого символа не оказалось
<code>char *strrchr(const char *s, int c)</code>	возвращает указатель на последнее вхождение символа <i>c</i> в строку <i>s</i> , либо NULL, если такого символа не оказалось
<code>char *strpbrk(const char *s, const char *t)</code>	возвращает указатель в строке <i>s</i> на первый символ, который совпал с одним из символов, входящих в <i>t</i> , либо NULL, если такового не оказалось
<code>char *strstr(const char *s, const char *t)</code>	возвращает указатель на первое вхождение строки <i>t</i> в строку <i>s</i> , либо NULL, если такой подстроки в <i>s</i> не оказалось
<code>char *strtok(char *s, const char *t)</code>	ищет в <i>s</i> лексему, ограниченную символами из <i>t</i> . Возвращает указатель на первый символ лексемы, либо NULL, если лексемы не существует

Рисунок 2 – иллюстрация списка основных функций

Прототип	Описание функции
double atof(const char *s)	переводит строку <i>s</i> в действительное число
int atoi(const char *s)	переводит строку <i>s</i> в целое число
long atol(const char *s)	переводит строку <i>s</i> в длинное целое число
double strtod(const char *s, char **endp)	преобразует первые символы строки <i>s</i> в тип double, игнорируя начальные символы-разделители; если endp не NULL, то указателю *endp присваивается адрес символа, который является первым символом строки-остатка после преобразования
long strtol(const char *s, char **endp, int base)	преобразует первые символы строки <i>s</i> в тип long, игнорируя начальные символы-разделители; указателю *endp присваивается адрес символа, который является первым символом строки-остатка после преобразования; base – основание, по которому производится преобразование
unsigned long strtoul(const char *s, char **endp, int base)	работает так же, как и strtol(), только возвращает результат типа unsigned long

Рисунок 3 – иллюстрация списка основных функций библиотеки <stdlib.h>

Приложение А

(обязательное)

Код программы задания 1

```
#include<iostream>
#include<ctype.h>
#include<cstring>
#include<cstdlib>
using namespace std;
char str[100];
char* nextW = NULL;
int main()
{
    /*напечатать все слова, не содержащие ни одной большой латинской буквы*/
    cin.getline(str, 100);
    cout << str << endl;
    char* words = strtok_s(str, "_.,;\\n\\t!?", &nextW);
    //cout << words;
    while (words != NULL) {
        bool b = true;
        for (int i = 0; i < strlen(words); i++) {
            if (isupper(words[i])) {
                b = false;
            }
        }
        if (b) {
            cout << words << ' ';
        }
        words = strtok_s(NULL, "_.,;\\n\\t!?", &nextW);
    }
}
```

Приложение Б

(обязательное)

Код программы задания 2

```
#include<iostream>
#include<ctype.h>
#include<cstring>
#include<cstdlib>
using namespace std;
char str[100];
char* nextW = NULL;
int main()
{
    /*напечатать все слова, не содержащие ни одной большой латинской буквы*/
    cin.getline(str, 100);
    cout << str << endl;
    int spaces[10];
    int countS = 0;
    for (int i = 0; i < strlen(str); i++) {
        if (isspace(str[i])) {
            spaces[countS] = i;
            countS++;
        }
    }
    for (int i = 0; i < countS; i++) {
        cout << spaces[i] << ' ';
    }
}
```


Приложение В

(обязательное)

Код программы задания 3

```
#include<iostream>
#include<ctype.h>
#include<cstring>
#include<cstdlib>
using namespace std;
char str[100];
char* nextW = NULL;
int main()
{
    /*Дан текст. Посчитать кол-во цифр*/
    cin.getline(str, 100);
    cout << str << endl;
    int countN = 0;
    for (int i = 0; i < strlen(str); i++) {
        if (isdigit(str[i])) {
            countN++;
        }
    }
    cout << countN;
}
```