

Лабораторная работа №14

Обработка строк с использованием специальных функций.

Оглавление

Цель	1
Инструкция:	2
Задания для самостоятельного выполнения:	4
Домашнее задание:.....	4
БЛОК А.....	4
БЛОК В	4
БЛОК С	4
Требования к оформлению программ:	5
Контрольные вопросы:	5

Цель

Научиться обрабатывать строки с использованием специальных функций.

Инструкция:

Пример 1: В строке заменить все вхождения подстроки “yes” на “no”.

```
#include<iostream>
#include<string.h>

using namespace std;
int main()
{
    char s[100]={"Yes no yes yeS yeS YES";      //первоначальная строка
    char s1[]="yes";      //искомая подстрока
    char s2[]="no";      //строка на замену
    char *p,*st;          //вспомогательные указатели
    int n,n1,n2;          //р - указатель на начало найденной подстроки
                           //st - еще не обработанная часть строки
                           // длины строк s,s1,s2 соответственно

    n=strlen(s);

    cout<<"Original string: \t"<<s<<endl;
    cout<<"String length: "<<n<<endl;

    n=strlen(s);
    n1=strlen(s1);
    n2=strlen(s2);

    //если в строке есть хотя бы один символ, то начинаем ее обработку
    if (n>0)
    {
        st=s;
        p=strstr(st,s1);      //ищем подстроку

        while(p!=NULL)          //выполняем пока нужные подстроки есть
        {
            strcpy(p,p+(n1-n2)); //строка "yes" длиннее, чем "no".
            strncpy(p,s2,n2);   //поэтому укорачиваем строку на один символ
                           //и в найденное место копируем строку "no"
            st=p+n2;
            p=strstr(st,s1);    //сокращаем "необработанную" часть строки
                           // заново ищем подстроку
        }

        cout<<"Modified string: \t"<<s<<endl; //выводим результат
        n=strlen(s);
        cout<<"String length: "<<n<<endl;      //и длину строки после обработки
    }
    else cout<<"Error: zero string length!";
}

return 0;
```

Пример 2: В тексте найти и вывести на экран все слова, в которых есть цифры.

```

#include<iostream>
#include<iomanip>
#include<string.h>
#include<ctype.h>

using namespace std;
bool chek_string(char* s,int& num)
{
    int i=0;
    while(s[i]!='\0')
    {
        if(isdigit(s[i]))
        {
            num=i+1;
            return true;
        }
        i++;
    }
    return false;
}

int main()
{
    char str[100] = "Do you like programming? Yes, I like programming very much! ";
    char *p;                                //указатель на новое слово
    int num;

    cout<<"Original check string: "<<endl<<endl;
    cout<<str<<endl<<endl;

    p=strtok(str," .!?,;:") ;      //поиск слова

    cout<<"Mistakes in words: "<<endl;
    while(p)
    {
        if(chek_string(p,num))
            cout<<setw(20)<<p<<setw(25)
            <<" -> Mistake in the "
            <<num<<" symbol"<<endl;
        p=strtok(NULL," .!?,;:");
    }

    return 0;
}

```

Пример 3: Дано предложение, начинающееся с пробелов. Получить новое, удалив все начальные пробелы.

```

#include<iostream>
#include<string.h>
#include<ctype.h>
using namespace std;

int main()
{
    char str[100] = "Hello, my friend:)";
    char *s,*st;

    int i=0;

    cout<<str<<endl;

    while(isspace(str[i]) && str[i]!='\0')
    {
        i++;
    }
    strcpy(str,str+i);
    cout<<str<<endl;

    return 0;
}

```

Задания для самостоятельного выполнения:

1. В файле имеется некоторый текст. Выведите на экран все слова этого текста, указывая их длину. Учитывайте знаки пунктуации.
2. Дано предложение, заканчивающееся пробелами. Получить новое, удалив все пробелы из конца предложения.

Домашнее задание:

БЛОК А

Задание на отметку «удовлетворительно»

1. В тексте найти и вывести на экран все слова, в которых есть цифры. Определить сколько цифр в каждом такой найденном «неправильном» слове и что это за цифры.
2. В прочитанном из файла тексте заменить все вхождения подстроки «те» на «you» .

БЛОК В

Задания на отметку «хорошо»

Выполнить задание из ***Лабораторной работы № 13 БЛОК В***, используя специальные функции для обработки строк.

БЛОК С

Задания на отметку «отлично»

Выполнить задание из ***Лабораторной работы № 13 БЛОК С***, используя специальные функции для обработки строк.

Задание:

Требования к оформлению программ:

1. **Содержание.** Программа должна делать то, что предусмотрено заданием. Не надо выполнять лишних действий, заданием не предусмотренных.
 2. **Спецификация.** В преамбуле программы в комментариях указывать сведения:
 - Кто выполнил.
 - Что делает программа (кратко).
 - Что на входе (имена входных файлов и т.д.).
 - Что на выходе (что является результатом работы программы?).
 3. **Ввод и вывод**
 - Приглашения к вводу (например, сколько чисел, какого типа и через какой разделитель нужно вводить).
 - Контрольный вывод (все введенные данные выводить на экран, и только после этого выполнять необходимые вычисления.)
 - «Защита от дурака». Проверять вводимые данные на корректность. Например, если необходимо считать количество чего – то, то эта величина не может быть отрицательной и т.д.
 4. **Структура кода.** Набираемый код должен быть хорошо структурированным.
Использовать:
 - Отступы.
 - Комментарии – поясняют решение программы.
 - Осмысленные названия переменных.
 - Пояснения о назначении переменных в комментариях (кроме счетчиков).
 5. **Декомпозиция кода**
 - Функциональная. Программу оформлять с помощью функций.
 6. **Многофайловые проекты**
 - Все проекты должны состоять минимум из двух модулей: главного и подключаемого.
- Главный модуль **main.cpp**. В нем оставить только функцию main, в которой вызывать функции, описанные в других модулях.
- Модуль с описанием пользовательских функций **file.cpp** и **file.h**. В этот модуль перенести определение всех функций, необходимых для выполнения задания.

Контрольные вопросы:

Заполнить таблицу о стандартных функциях обработки строк. Для каждой функции указать ее назначение; заголовочный файл, в котором она определена; прототип функции, описание всех аргументов; и пример использования.

Пример заполнения таблицы:

Функция нахождения длины строки	
Заголовочный файл:	string.h
Прототип:	int strlen (char *str) str – входная строка функция возвращает количество символов в строке str
Пример:	<pre>#include <stdio.h> #include <string.h> int main() { char str[50] = "Do you like programming? I love programming!"; puts(str); printf("String line: %d\n", strlen(str)); return 0; }</pre>

Список функций:

- | | | |
|------------------------------------|---|--|
| 1. strlen | 5. strcmp
strncmp | 9. atoi
atof
itoa |
| 2. strtok | 6. strcpy
strncpy | 10. strrev |
| 3. strcat
strncat | 7. islower
isupper
isalpha | 11. strupr
strlwr |
| 4. strstr | 8. isdigit
isspace
ispunct | |