Лабораторная работа №11.

**Тема работы:** Разработка и отладка алгоритмов и программ обработки строковых данных

**Цель работы:** Сформировать умения разрабатывать и тестировать программы обработки строковых данных

**Оснащение рабочего места:** ПК, VS, методические указания для проведения лабораторных работ, индивидуальные задания на ЛР

**Краткие теоретические сведения**

Строки в С++ позволяют работать с символьными данными. С помощью строк возможно осуществить чтение с клавиатуры текста, его обработка и вывод.

В С++ существует 2 типа строк:

1. Массив переменных типа char, заканчивающийся нуль-терминатором \0. Символьные строки состоят из набора символьных констант заключённых в двойные кавычки. При объявлении строкового массива необходимо учитывать наличие в конце строки нуль-терминатора, и отводить дополнительный байт под него.

char string[10]; // string – имя строковой переменной, 10 – размер массива, (в строке может поместиться 9 символов, последнее место отводится под нуль-терминатор)

1. Специальный класс string

Для его работы необходимо в начале программы подключить заголовочный файл string: *#include <string>*

Для создания строки необходимо в начале программы написать

*using namespace std;*

Теперь чтоб создать строку достаточно написать: *string s;*

Для записи в строку можно использовать оператор =

s="Hello";

Доступ к i-му элементу строки **s** типа string осуществляется стандартным образом s[i]. Над строками типа string определены следующие операции:

* + присваивания, например s1=s2;
  + объединения строк (s1+=s2 или s1=s1+s2) — добавляет к строке s1 строку s2, результат храниться в строке s1, пример объединения строк:
  + сравнения строк: s1=s2, s1!=s2, s1<s2, s1>s2, s1<=s2, s1>=s2 — результатом будет логическое значение.

Существует множество функций для работы со строками (таблица 7). Таблица 7 – Функции для работы со строками

|  |  |
| --- | --- |
| Функция | Объяснение |
| s.append(str) | добавляет в конец строки строку str |
| s.assign(str) | присваивает строке s значение строки str |
| s.clear() | отчищает строку, т.е. удаляет все элементы в ней |
| s.compare(str) | сравнивает строку s со строкой str и возвращает 0 в случае  совпадение |
| s.copy(C, I, N) | копирует из строки s в строку C(может быть как строка  типа string, так и строка типа char) I символов, начиная с N-го символа |
| bool b=s.empty() | если строка пуста, возвращает true, иначе false |
| s.erase(I,N) | удаляет N элементов с I-го символа |
| s.find(str,I) | ищет строку str начиная с I- го символа |
| s.insert(pos, s1) | вставляет строку **s1** в строку **s**, начиная с позиции **pos** |
| int len=s.length() | записывает в len длину строки |
| s.push\_back(symbol) | добавляет в конец строки символ |
| s.replace(index,  n,str) | берет n первых символов из str и заменяет символы строки  s на них, начиная с позиции index |
| str=s.substr(n,m) | возвращает m символов начиная с позиции n |
| s.swap(str) | меняет содержимое s и str местами. |
| s.size() | возвращает число элементов в строке. |

Функции для работы со строками, прототипы которых описаны в заголовочном файле string.h:

1. Сравнение строк. Для сравнения строк используются функции strcmp и strncmp. Функция

int strcmp ( const char \*str1, const char \*str2);

лексикографически сравнивает строки str1, str2 и возвращает –1, 0 или 1, если строка str1 соответственно меньше, равна или больше строки str2.

Функция

int strncmp ( const char \*str1, const char \*str2, size\_t n);

лексикографически сравнивает не более чем n первых символов из строк str1 и str2. Функция возвращает –1, 0 или 1, если первые n символов из строки str1 соответственно меньше, равны или больше первых n символов из строки str2.

**Порядок выполнения работы**

1. Ознакомиться с краткими теоретическими сведениями
2. Ознакомиться с поставленной задачей на ЛР

**Задания (по вариантам)**

Дана последовательность символов S1, …, SN. Группы символов, разделенные пробелом (одним или несколькими) и не содержащие пробелов внутри себя, будем называть словами.

1. Определить число символов в самом длинном слове строки. Слова отделяются знаком “/”.
2. В произвольном тексте выделить и отпечатать слова, начинающиеся с буквы А.
3. В произвольном тексте вставить между первым и вторым словом новое слово.
4. В произвольном тексте найти и отпечатать слова, содержащие букву Е.
5. Отпечатать второе и третье слова произвольного текста.
6. В произвольном тексте вставить между вторым и третьим словом новое слово.
7. В произвольном тексте найти и отпечатать все слова длиной 5 символов.
8. В произвольном тексте найти самое короткое слово.
9. В последовательности из 10 пятибуквенных слов найти и поменять местами пару слов, у которых первые три буквы одного совпадают с последними тремя буквами другого.
10. Упорядочить в алфавитном порядке последовательность из 10 пятибуквенных слов.
11. В строке из 50 символов отдельные слова разделены пробелом. Упорядочить строку так, чтобы каждое следующее слово было не короче предыдущего.
12. Расположить слова строки в порядке, обратным исходному.
13. Подсчитать количество букв ‘a’ в последнем слове строки.
14. Найти количество слов, у которых первый и последний символы совпадают между собой.
15. Исключить из строки слова, расположенные между скобками ( , ). Сами скобки должны быть исключены.
16. В произвольном тексте найти и отпечатать слова, содержащие букву А.
17. Решить программным способом поставленную задачу
18. Составить отчёт по выполненной ЛР

**Содержание отчета:**

1. Титульный лист (см. образец отчёта)
2. Наименование и цель работы.
3. Задание на ЛР
4. Скопированный код из VS.
5. Скриншот выполненной программы
6. Ответы на контрольные вопросы.
7. Выводы по работе.

**Контрольные вопросы**

1. Строки в языке С++: понятие строки, описание строк в программе, обращение к элементам строки.
2. Способы ввода строк в С++.
3. Способы вывода строк в С++.
4. Способы инициализации строк (задание значений в программе).
5. Стандартные функции для обработки строк.

**Литература**

Бансила М. Решение задач на современном С++ : Пер. с анг.