**Лабораторная работа № 13. Обработка символьной информации**

Строка символов хранится в памяти как *массив*. Каждый элемент массива содержит один символ, при этом последним символом является '**\0**'. Поэтому при объявлении массива надо указывать размерность на единицу больше, чем количество символов. Имя массива без индекса является **указателем-константой**.

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1.Выполнить прокрутки программ и записать условия.  Опробовать программы с различным текстом. | **#include <iostream>**  **void main()**  **{**  **char str[] = "Text";**  **int count = 0;**  **char \*pstr;**  **pstr = &str[0];**  **if (str)**  **while (\*pstr++)**  **++count;**  **std::cout<<count;**  **}**  **#include <iostream>**  **void main()**  **{**  **int num = 5;**  **char symb, new\_symbol = ' ';**  **char\* pc; pc = &symb;**  **\*pc = num + '0';**  **std::cout << \*pc << ' ';**  **std::cout << symb << ' ';**  **if (symb >= '0' && symb <= '9')**  **num = symb - '0';**  **std::cout << num << ' ';**  **symb = 'b';**  **if (symb >= 'a' && symb <= 'z')**  **new\_symbol = symb - 'a' + 'A';**  **std::cout << new\_symbol << ' ';**  **}**    Программа подсчитывает количество символов в строке "Text" (без учета нулевого символа '\0').  Результат будет равен 4, так как строка "Text" содержит 4 символа.  Первая часть кода преобразует число num в символ  и выводит его дважды.  Затем программа проверяет, является ли symb цифрой,  и если да, преобразует его обратно в число и выводит.  В последней части кода symb устанавливается в 'b',  и если он является строчной буквой, преобразуется  в прописную букву и выводится. |
| 2. Выполнить программу, приведенную в правой части.  Объяснить принцип использования *функций стандартной библиотеки*. | Пример. Имеется адрес файла в сети, например, **http://belstu.by/p1/p2/file1.htm**.Определить имя последней папки (каталога).   |  |  | | --- | --- | | **#include <cstring>**  **#include <iostream>**  **using namespace std;**  **void main()**  **{**  **char str[50], s[20];**  **int i, j, i1, i2;**  **cout<<"Input string:";**  **cin>> str;**  **for(i = strlen(str); i >= 0; i--)**  **if (str[i] == '\/') break;**  **i2 = i;**  **for (j = (i2 - 1); j >= 0; j--)**  **if (str[j] == '\/') break;**  **i1 = j;**  **if (i2 == i1)**  **printf(" '\/' ");**  **else**  **{ strncpy\_s(s, &str[i1+1], i2-i1-1);**  **s[i2 - i1 - 1] = 0;**  **cout<<"katalog:"<< s<<"\n";**  **}**  **}** | **char \*strncpy(char \*s1, const char \*s2, int kol);**  Копирует kol символов строки s2 в строку s1  **int strlen (const char \*str);**  Вычисляет длину строки str | |
| 3. Изучить способы работы с символьными массивами, выполнив программу в правой части. | Пример. Из предложения удалить все символы, совпадающие с символом, введенным с клавиатуры.  **void main()**  **{ char s, t[20];**  **int i, j, n;**  **cout<<"Input string: ";**  **gets\_s(t);**  **cout<<"Input symbol: ";**  **cin>>s;**  **n = strlen(t);**  **for(i = 0; i <= n; i++)**  **if (t[i] == s)**  **for(j = i; j <= n; j++)**  **t[i] = t[i + 1];**  **for(i = 0; i < n; i++)**  **cout << t[i];**  **}** |
| 4. Выполнить программу в правой части.  Закомментировать строку **cin.ignore…**  Объяснить возникшие проблемы при выполнении программы. | **void main()**  **{ char t[3][20];**  **int i, n;**  **cout<<"Input size (<=3): ";**  **cin>>n;**  **cin.ignore(cin.rdbuf()->in\_avail());**  **for(i = 0; i < n; i++)**  **{ cout<<"Input string: ";**  **gets\_s(t[i]);**  **}**  **for(i = 0; i < n; i++)**  **puts(t[i]);**  **}**    Закомментирование строки cin.ignore(cin.rdbuf()->in\_avail()); приводит к проблемам с буфером ввода, что вызывает некорректное поведение программы при вводе строк. |
| 5. В правой части записаны два варианта решения задачи. Выполнить программы и объяснить различия между ними.  Внести изменения в программу с тем, чтобы проверялось не только количество скобок, но и правильность их расстановки.  6. Выполнить задания из таблицы ниже, используя ***индексы*** для доступа к элементам массивов в первой программе и ***указатели*** − во второй программе.  Вариант 8   1. Написать программу, которая осуществляет сравнение двух строк и выводит сообщение о том, какие символы совпадают.   2. Дана строка, состоящая из групп нулей и единиц. Найти группы с четным количеством символов | Пример проверки соответствия количества открывающих и закрывающих круглых скобок в строке.   |  |  | | --- | --- | | **#include <stdio.h>**  **void main()**  **{**  **char s[256];**  **int i, count;**  **puts("Enter string: ");**  **gets\_s(s);**  **for(count=i=0; s[i] != 0; i++)**  **{**  **if (s[i] == '(') count++;**  **if (s[i] == ')') count--;**  **}**  **if (!count)**  **puts("Ok\n");**  **else**  **puts("Not Ok\n");**  **}** | **#include <stdio.h>**  **void main()**  **{**  **char s[256];**  **int count;**  **char \*ps;**  **puts("Enter string: ");**  **gets\_s(s);**  **for(count = 0, ps = s; \*ps != 0; ps++)**  **{**  **if (\*ps == '(') count++;**  **if (\*ps == ')') count--;**  **}**  **if (count == 0)**  **puts("Ok\n");**  **else**  **puts("Not Ok\n");**  **}** |     Различие в том, что в первой программе доступ к элементам осуществляется через индекс s[i], а во второй доступ к элементам осуществляется через разыменование указателя \*ps.    **#include <stdio.h>**  **int main()**  **{**  **char s[256];**  **int i, count;**  **puts("Enter string: ");**  **gets\_s(s);**  **for (count = i = 0; s[i] != 0; i++) {**  **if (s[i] == '(') {**  **count++;**  **}**  **if (s[i] == ')') {**  **count--;**  **if (count < 0) {**  **break;**  **}**  **}**  **}**  **if (count == 0 && s[i] == 0) {**  **puts("Ok\n");**  **}**  **else {**  **puts("Not Ok\n");**  **}**  **return 0;**  **}**    **#include <iostream>**  **#include <string>**  **using namespace std;**  **int main() {**  **setlocale(LC\_ALL, "ru");**  **string str1, str2;**  **cout << "Введите первую строку: ";**  **cin >> str1;**  **cout << "Введите вторую строку: ";**  **cin >> str2;**  **int minLength = min(str1.length(), str2.length()); // Находим минимальную длину строк**  **cout << "Совпадающие символы: ";**  **for (int i = 0; i < minLength; i++) {**  **if (str1[i] == str2[i]) {**  **cout << str1[i] << " ";**  **}**  **}**  **cout << endl;**  **}**   В этой программе используется доступ к элементам строк через индексы str1[i] и str2[i].   Мы находим минимальную длину строк с помощью min и сравниваем символы по индексам.   Если символы совпадают, они выводятся на экран.     В этой программе используется указатель p1 и p2, которые указывают на первые символы строк.  **#include <iostream>**  **#include <string>**  **using namespace std;**  **int main() {**  **setlocale(LC\_ALL, "ru");**  **string str1, str2;**  **cout << "Введите первую строку: ";**  **cin >> str1;**  **cout << "Введите вторую строку: ";**  **cin >> str2;**  **const char\* p1 = str1.c\_str();**  **const char\* p2 = str2.c\_str();**  **while (\*p1 && \*p2) {**  **if (\*p1 == \*p2) {**  **cout << \*p1 << " ";**  **}**  **p1++;**  **p2++;**  **}**  **cout << endl;**  **}**   Указатели инкрементируются, чтобы перейти к следующему символу.   Если символы, на которые указывают указатели, совпадают, выводим их.    **#include <iostream>**  **#include <string>**  **using namespace std;**  **int main() {**  **setlocale(LC\_ALL, "ru");**  **string binaryStr;**  **cout << "Введите строку из нулей и единиц: ";**  **cin >> binaryStr;**  **int count = 1;**  **for (int i = 1; i < binaryStr.length(); i++) {**  **if (binaryStr[i] == binaryStr[i - 1]) {**  **count++;**  **} else {**  **if (count % 2 == 0) {**  **cout << "Группа с четным количеством символов: ";**  **for (int j = i - count; j < i; j++) {**  **cout << binaryStr[j];**  **}**  **cout << endl;**  **}**  **count = 1;**  **}**  **}**  **// Для последней группы, если она была четной**  **if (count % 2 == 0) {**  **cout << "Группа с четным количеством символов: ";**  **for (int j = binaryStr.length() - count; j < binaryStr.length(); j++) {**  **cout << binaryStr[j];**  **}**  **cout << endl;**  **}**  **}**   В этой программе мы проходим по строке, считая длину группы символов (нулей или единиц), и если длина группы четная, выводим ее.   Используем индексы для доступа к символам строки.    **#include <iostream>**  **#include <string>**  **using namespace std;**  **int main(){**  **setlocale(LC\_ALL, "ru");**  **string binaryStr;**  **cout << "Введите строку из нулей и единиц: ";**  **cin >> binaryStr;**  **const char\* p = binaryStr.c\_str();**  **const char\* start = p;**  **int count = 1;**  **while (\*p) {**  **if (\*p == \*(p - 1)) {**  **count++;**  **}**  **else {**  **if (count % 2 == 0) {**  **cout << "Группа с четным количеством символов: ";**  **for (const char\* temp = start; temp < p; temp++) {**  **cout << \*temp;**  **}**  **cout << endl;**  **}**  **count = 1;**  **start = p;**  **}**  **p++;**  **}**  **// Для последней группы**  **if (count % 2 == 0) {**  **cout << "Группа с четным количеством символов: ";**  **for (const char\* temp = start; temp < p; temp++) {**  **cout << \*temp;**  **}**  **cout << endl;**  **}**  **}**   В этой программе мы используем указатели для прохождения по строке.   Указатель p используется для перебора символов строки, а start сохраняет начало текущей группы символов.   При смене символа проверяется, была ли группа четной, и если да, она выводится. |