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

|  |  |  |
| --- | --- | --- |
| Задание | | Решение |
| 1. Изучить способы преобразования символов, выполнив программы в правой части.  Выполнить прокрутки программ и записать условия.  Опробовать программы с различным текстом. | | **#include <iostream>**  **void main()**  **{ int num = 5;**  **char symb, new\_symbol = ' ';**  **symb = num + '0';**  **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<<' ';**  **}**    #include <iostream>  void main()  {  int num = 5;  char symb, new\_symbol = ' ';  char\* pc; pc = &symb;  \*pc = num + '0'; std::cout << \*pc << ' ';  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 << ' ';  }    Результат тот же, только symb изменяется с помощью указателя     1. num = 5; new\_symbol = ' '; symb = num + '0' = '5'; вывод symb = '5'; 2. symb = '5'; symb >= '0' && symb <= '9'; num = symb - '0' = 5; вывод num = 5; 3. symb = 'b'; symb >= 'a' && symb <= 'z'; new\_symbol = symb - 'a' + 'A' = 'B'; вывод new\_symbol = 'B'; 4. Выполнение программы закончено.   **#include <iostream>**  **void main()**  **{**  **char str[] = "Text";**  **int count = 0;**  **char \*pstr;**  **pstr = &str[0];**  **if (str)**  **while (\*pstr++)**  **++count;**  **std::cout<<count;**  **}**       1. str = "Text"; count = 0; pstr = &str[0]; 2. str истина; \*pstr = 'T'; pstr++; count = count + 1 = 1; 3. \*pstr = 'e'; pstr++; count = count + 1 = 2; 4. \*pstr = 'x'; pstr++; count = count + 1 = 3; 5. \*pstr = 't'; pstr++; count = count + 1 = 4; 6. \*pstr = '\0'; условие \*pstr ложь; выход из цикла; 7. вывод count = 4.   Условие:  Найти и вывести количество символов в строке |
| 2. Выполнить программу, приведенную в правой части.  Объяснить принцип использования *функций стандартной библиотеки.* | | **#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";**  **}**  **}** |
| 3. Изучить способы работы с символьными массивами, выполнив программу в правой части. | | #include <iostream>  using namespace std;  void main()  {  char s, t[200];  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]);**  **}**    Если закомментировать cin.ignore |
| 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 (первой в тексте должна быть открывающая скобка). | | **#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");**  **}**    Вторая программа использует указатели для обхода массива и if сделан чуть более читаемым  #include <stdio.h>  void main()  {  char s[256];  int i, count;  puts("Enter string: ");  gets\_s(s);  if (s[0] != '(') {  puts("Not Ok\n");  return;  }  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");  } |
| 6. Выполнить задания из таблицы ниже, используя ***индексы*** для доступа к элементам массивов в первой программе и ***указатели*** − во второй программе.  При написании программне использовать стандартные функции для строк символов. | | |
| №Варианта  10 | | |
| Задание | Код/Результат | |
| 1. Написать программу, которая записывает строку в обратном порядке. | #include <iostream>  using namespace std;  int main() {  char a[100];  cin >> a;  int i = 99;  while (a[i] != '\0') {  i--;  }  while (i > -1) {  cout << a[i];  i--;  }  return 0;  } | |
| 2. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Разбить исходную строку на две подстроки. Первая подстрока должна иметь длину **k** символов (если на **k**-ю позицию попадает слово, то его следует отнести ко второй строке). | #include <iostream>  using namespace std;  int main() {  char a[100];  char b[100];  char c[100];  cin.getline(a, 100);  cin.ignore(cin.rdbuf()->in\_avail());  int k;  cin >> k;  int starti = 0;  int bi = 0, ci = 0;  int i = 0;  while (\*(a+i-1) != '\0') {  if (i <= k) {  if (\*(a + i) != ' ') {  i++;  }  else {  for (int j = starti; j <= i; j++) {  \*(b + bi + (j - starti)) = \*(a + j);  }  bi += i - starti + 1;  i++;  starti = i;  }  }  else {  \*(b + bi) = '\0';  if (\*(a + i) == ' ' || \*(a + i) == '\0') {  for (int j = starti; j <= i; j++) {  \*(c + ci + (j - starti)) = \*(a + j);  }  ci += i - starti + 1;  i++;  starti = i;  }  else {  i++;  }  }  }  \*(c +ci) = '\0';  for (char\* i = b; \*i != '\0'; i++) {  cout << \*i;  }  cout << '\n';  for (char\* i = c; \*i != '\0'; i++) {  cout << \*i;  }  return 0;  } | |

|  |  |
| --- | --- |
| Допы | |
| Задание | Код/результат |
| 1.В заданной последовательности слов найти все слова, имеющие заданное окончание | setlocale(LC\_ALL, "rus");  SetConsoleCP(1251);  SetConsoleOutputCP(1251);  char str[1000];  cin.getline(str, 1000);  char en[256];  cin.ignore(cin.rdbuf()->in\_avail());  cin >> en;  int i = 0, j = 0;  int start = 0;  while (str[i] != '\0') {//не конец строки  if (str[i] == ' ') {//кончилось слова  if (en[j] == '\0') {//прошли всё окончание  for (int k = start; k < i; k++) {//выводим со старта  cout << str[k];  }  cout << ' ';  }  j = 0;//сбрасываем окончание  start = i + 1;//ставим начало нового слова  }  if (str[i] == en[j]) {  j++;//сравниваем со следующей буквой  }  else {  j = 0;//считаем сначала окончание  }  i++;//следующая буква в мейн строке  }  if (en[j] == '\0') {//проверка последнего слова  for (int k = start; k < i; k++) {  cout << str[k];  }  cout << ' ';  }  break; |
| 3. Из заданного предложения удалить те слова, которые уже встречались в предложении раньше. | setlocale(LC\_ALL, "rus");  SetConsoleCP(1251);  SetConsoleOutputCP(1251);  char str[1000];  cin.getline(str, 1000);  char word[256];  int i = 0, j = 0;  int start = 0;  while (str[i] != '\0') {  word[j] = str[i];//добавляем следующую букву в слово  if (str[i] == ' ') {//слово кончилось  j = 0;//начинаем смотреть слово с нуля  bool flag = true;//флаг что мы не нашли такое же слово(по умолчанию тру)  for (int k = 0; k < start; k++) {//смотрим всю строку с начала до начала текущего слова  if (word[j] == ' ') {//конец слова  if (str[k] == ' ') {// и конец строки  flag = false;//нашли такое же слово, значит не подходит выводить не будем  break;//ливаем с цикла на это слово уже все равно  }  else {//нашли слово внутри другого слова, скипаем текущее слово  while (str[k] != ' ' && str[k] != '\0') {//скипаем текущее слово до пробела или конца строки( с отсальным разберётся условие цикла)  k++;  }  }  }  if (str[k] == word[j]) {//буква совпала проверяем следующую  j++;  }  else {//буква не совпала слово полностью не подходит смотрим следующее  j = 0;  while (str[k] != ' ' && str[k] != '\0') {  k++;  }  }  }  if (flag) {//если флаг не стал фолс значит не нашли такое же слово  for (int k = 0; k < i - start; k++) {//выводим слово  cout << word[k];  }  cout << ' ';//пробел :)  }  start = i + 1;//переставляем начало для нового слова  j = 0;//начинаем записывать новое слово с нуля  i++;  }  else {  j++;  i++;  }  }  word[j] = ' ';//ставим в конец слова пробел, чтобы условия работали и код переписывать не пришлось  j = 0;//проверяем последнее слово, см комменты сверху  bool flag = true;  for (int k = 0; k < start; k++) {  if (word[j] == ' ') {  if (str[k] == ' ') {  flag = false;  break;  }  else {  while (str[k] != ' ' && str[k] != '\0') {  k++;  }  }  }  if (str[k] == word[j]) {  j++;  }  else {  while (str[k] != ' ' && str[k] != '\0') {  k++;  }  }  }  if (flag) {  for (int k = 0; k < i - start; k++) {  cout << word[k];  }  cout << ' ';  } |
| 4. Отредактировать заданное предложение, удаляя из него все слова с нечетными номерами и переворачивая слова с четными номерами. Пример: HOW DO YOU DO преобразовать в OD OD. | const int N = 100;  char str[N];  cout << "Enter string: ";  cin.getline(str, N);  unsigned int m;  m = strlen(str);  for (int i = m - 1; i >= 0; i--)  {  cout << str[i];  if (str[i] == ' ')  {  i--;  while (str[i] != ' ')  {  i--;  }  }  } |