Задание 1

|  |  |  |
| --- | --- | --- |
| a)   |  | | --- | | 1. Есть числовая переменная num, равная 5. Преобразуйте её в символ цифры. Затем проверьте, является ли этот символ цифрой, и преобразуйте его обратно в число. 2. Есть символ 'b'. Если он является строчной буквой, преобразуйте его в заглавную букву. |   **#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<<' ';**  **}**    Прокрутка:  num=5;  symb, new\_symbol = ' ';  symb=num+ '0';  Вывод symb  symb>='0' и symb<='9'; num=symb-'0';  Вывод num ' '  symb='b';  symb >= 'a' и symb <= 'z'; new\_symbol = symb - 'a' + 'A';  Вывод new\_symbol ' '  Выполнение программы закончено.  Изменённый текст    б)   |  | | --- | | Этот код подсчитывает длину строки (количество символов до \0) с помощью указателя и выводит её. Результат для строки "Text": 4. |   #include <iostream>  void main()  {  char str[] = "Text";  int count = 0;  char\* pstr;  pstr = &str[0];  if (str)  while (\*pstr++)  ++count;  std::cout << count;  }    Прокрутка:  str[] = "Text";  count = 0;  pstr = &str[0];  str[0]=’T’  \*pstr=’T’; count++;(1)  \*pstr=’e’; count++;(2)  \*pstr=’x’; count++;(3)  \*pstr=’t’; count++;(4)  \*pstr=’/0’ выход из цикла  Вывод count(4)  Выполнение программы закончено. |

Задание 2

|  |
| --- |
| #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";  }  } |

Задание 3

|  |
| --- |
| **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];**  **}**  Эта программа работает, если символ встречается один раз      Доработанный вариант:  #include <cstring>  #include <iostream>  using namespace std;  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[j] = t[j + 1];  n--;  i--;  }  for (i = 0; i < n; i++)  cout << t[i];  } |

Задание 4

|  |
| --- |
| #include <cstring>  #include <iostream>  using namespace std;  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]);  }    Изменённый вариант  #include <cstring>  #include <iostream>  using namespace std;  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]);  }    После ввода числа n через cin >> n;, оставшийся символ новой строки (\n) не будет удалён из буфера ввода.  Когда программа далее перейдёт к чтению строк через gets\_s(t[i]);, она сразу же прочитает этот символ новой строки, так как буфер ввода ещё не был очищен.  Это приведёт к тому, что первое выполнение gets\_s сразу же завершится, не давая пользователю ввести строку. |

Задание 5

|  |
| --- |
| Изменённый вариант 1  #include <stdio.h>  void main()  {  char s[256];  int i,count;  bool isValid = true;  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)  {  isValid = false;  break;  }  }  }  if (count!=0)  {  isValid = false;  }  if (isValid)  puts("Ok\n");  else  puts("Not Ok\n");  }      Вторая программа, такая же как и предыдущая, но с использованием указателя для обхода строки вместо обычного индекса.  #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");  } |

Вариант 12

|  |
| --- |
| Зашифровать введенную с клавиатуры строку, поменяв местами первый символ со вторым, третий с четвертым и т. д. |

|  |
| --- |
| #include <stdio.h>  #include<iostream>  using namespace std;  void main()  {  char s[256];  int i, count;  puts("Enter string: ");  gets\_s(s);  for (int i = 0; s[i] != 0; i++)  {  if (i % 2 == 0)  {  if (s[i + 1] != 0)  {  char temp = s[i];  s[i] = s[i + 1];  s[i + 1] = temp;  }  }  }  for (int i = 0; s[i] != 0; i++)  {  cout << s[i];  }  cout << endl;  } |

|  |
| --- |
| Дана строка символов, состоящая из букв, цифр, запятых, точек, знаков « + » и «–».Найти количество запятых и точек. |

|  |
| --- |
| #include <stdio.h>  #include <iostream>  void main()  {  char s[256];  int counttoch,countzap;  char\* ps;  puts("Enter string: ");  gets\_s(s);  for (counttoch = 0, countzap =0, ps = s; \*ps != 0; ps++)  {  if (\*ps == '.') counttoch++;  if (\*ps == ',') countzap++;  }  if (counttoch == 0)  {  std::cout << "counttoch=0 и countzap=" << countzap;  }    else if(countzap == 0)  std::cout << "countzap=0 и counttoch=" << counttoch;  else  {  std::cout << "countzap="<< countzap <<" и counttoch = " << counttoch;  }  } |

Доп задания:

Задание 1

|  |
| --- |
| 1. В заданной последовательности слов найти все слова, имеющие заданное окончание. |

|  |
| --- |
| #include <iostream>  #include <string>  #include <vector>  using namespace std;  int main() {  string sentence;  string ending;  cout << "Enter a sentence: ";  getline(cin, sentence);  cout << "Enter the ending to search for: ";  getline(cin, ending);  // разделяем строку на слова  vector<string> words;  size\_t start = 0, end;  while ((end = sentence.find(' ', start)) != string::npos) {  words.push\_back(sentence.substr(start, end - start));  start = end + 1;  }  words.push\_back(sentence.substr(start));  // Ищем слова с заданным окончанием  vector<string> matchingWords;  for (const auto& word : words) {  if (word.ends\_with(ending)) {  matchingWords.push\_back(word);  }  }  // Выводим результат  if (matchingWords.empty()) {  cout << "No words found with the ending '" << ending << "'." << endl;  }  else {  cout << "Words ending with '" << ending << "': ";  for (const auto& word : matchingWords) {  cout << word << " ";  }  cout << endl;  }  return 0;  } |

Задание 3

|  |
| --- |
| 1. Из заданного предложения удалить те слова, которые уже встречались в предложении раньше. |

|  |
| --- |
| #include <iostream>  #include <string>  #include <vector>  #include <algorithm>  using namespace std;  int main() {  string sentence;  cout << "Enter a sentence: ";  getline(cin, sentence);  vector<string> words;  size\_t start = 0, end;  while ((end = sentence.find(' ', start)) != string::npos) {  words.push\_back(sentence.substr(start, end - start));  start = end + 1;  }  words.push\_back(sentence.substr(start));  // удаляем повторяющиеся слова  vector<string> uniqueWords;  for (const string& word : words) {  if (find(uniqueWords.begin(), uniqueWords.end(), word) == uniqueWords.end()) {  uniqueWords.push\_back(word);  }  }  // cтрока из уникальных слов  cout << "Resulting sentence: ";  for (size\_t i = 0; i < uniqueWords.size(); ++i) {  if (i > 0) cout << " ";  cout << uniqueWords[i];  }  cout << endl;  return 0;  } |

Задание 4

|  |
| --- |
| 1. 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. |

|  |
| --- |
| #include <iostream>  #include <string>  #include <vector>  #include <algorithm>  using namespace std;  string reverseWord(const string& word) {  string reversed = word;  reverse(reversed.begin(), reversed.end());  return reversed;  }  int main() {  string sentence;  cout << "Enter a sentence: ";  getline(cin, sentence);  vector<string> words;  size\_t start = 0, end;  while ((end = sentence.find(' ', start)) != string::npos) {  words.push\_back(sentence.substr(start, end - start));  start = end + 1;  }  words.push\_back(sentence.substr(start));  string result = "";  // Обрабатываем слова  for (size\_t i = 0; i < words.size(); ++i) {  if (i % 2 == 1) {  result += reverseWord(words[i]) + " ";  }  }  cout << "Edited sentence: " << result << endl;  return 0;  } |