Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 13

По дисциплине «Основы алгоритмизации и программирования»

На тему «Обработка символьной информации»

Выполнил:

Студент 1 курса 6 группы

Кучерук Николай Петрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Задание**

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **9** | 1. Написать программу, реализующую вставку подстроки **St** длиной **n1** в строку **S** с позиции номер **n2**.  2. Дана строка, состоящая из групп нулей и единиц. Подсчитать количество символов в самой длинной группе. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <string>//данная директива нужна для использования типа данных string  using namespace std;  int main() {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  string st, p ;//объявляем строки  int n1,n2;  cout << "Введите строку S: "; getline(cin, p);//при помощи функции getline мы можем считать строку вместе с пробелами  cout << "Введите длину подстроку St : "; cin >> n2;  cout << "Введите подстроку St : "; cin >> st;  if (n2 != st.size()) { cout<<"Неверный размер подстроки St"; return 0; }//проверяем соответствует ли размер подстроки с введённым параметром  cout << "Введите позицию вставки строку S: "; cin>>n1;  cout << "S=" << p << ", St=" << st << endl;//выводим строку и подстроку до преобразований  p.insert(n1-1, st);// при помощи функции insert вставляем в строку p строку st начиная с n1-1  cout << p;//выводим обновленную строку p  return 0;  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <string>//данная директива нужна для использования типа данных string  using namespace std;  int main() {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  char str[100];//используем массив символов для хранения информации и возможности обращаться к нему через указатели  int max\_g=0,count=1;//начинаем счёт с 1,чтобы учитывать первый элемент, стоящий в группе  cout << "Введите строку S: ";  cin>>str;  for (int i=0; \*(str+i)!=0; i++) {//двигаемся по циклу пока не встретим последний символ '\0'  if (\*(str + i) == \*(str + i - 1)) { count++; }//пока не встретится неодинаковый элемент, ведем подсчет кол-ва символов  else { max\_g = max(max\_g, count); count = 1; }//при окончании группы сравниваем ее размер с уже известным макс. размером и обновляем счётчик  }  cout << max\_g;//выводим ответ  return 0;  } |
| **Результат программы** |
|  |

Дополнительные задания

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.

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <string>//данная директива нужна для использовния типа данных string  using namespace std;  string reverse(string str) {//данная функция возвращает введенную строку в перевернутом виде  int len = str.length();  int n = len - 1;//т.к. в конце строки всегда есть '\0',то его не учитываем в смене  int i = 0;  while (i <= n) {  swap(str[i], str[n]);//используем функцию swap для смены значений каждого индекса  n = n - 1;  i = i + 1;  }  return str;  }  int main() {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  string str,rev;//используем массив символов для хранения информации и возможности обращаться к нему через указатели  int max\_g=0,count\_s=0,count\_p=1,size;//начинаем счёт с 1,чтобы учитывать первый элемент,стоящий в группе  cout << "Введите строку S:";  getline(cin,str);  str.insert(str.size()," ");//добовляем в конце пробел для возможности обработки последнего слова  for (int i=0; i < str.size(); i++) {  if (str[i] == ' ' and count\_p%2!=0) {//если слово нечетное,то мы его удаляем  str.erase(i - count\_s , count\_s);  count\_p == 1 ? str.erase(0, 1) : str.erase(0, 0);//после удаления первого слова удаляем еще и пробел между ним и вторым словом  i -= count\_s;//возрващаемся на то место в строке,где находились до обработки нечетного слова  count\_s = 0;  count\_p++;  }  else if (str[i] == ' ' and count\_p % 2 == 0) {//если слово четное,то переворачиваем его  count\_p++;  count\_p > 3 ? count\_s -= 1 : count\_s -= 0;//после второго четного слова нужно не учитывать стоящий перед ним пробел  rev = str.substr(i - count\_s, count\_s);//извлекаем необходимое слово из строки  rev = reverse(rev);//переворачиваем его с помощью функции reverse  str.erase(i - count\_s, count\_s);//удаляем старое слово  str.insert(i - count\_s, rev);//и вставляем новое  count\_s = 0;  }  count\_s ++;//считаем кол-во символов в слове  }  cout << str;//выводим ответ  return 0;  } |
| **Результат программы** |
|  |

3.Из заданного предложения удалить те слова, которые уже встречались в предложении раньше.

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <string>//данная директива нужна для использования типа данных string  using namespace std;  int main() {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  string str1, str2,FIND;//объявляем строки  int count\_s = 0,pos;  cout << "Введите строку S1:"; getline(cin, str1);//при помощи функции getline мы можем считать строку вместе с пробелами  str1.insert(str1.size(), " ");//добавляем в конце пробел для возможности обработки последнего слова  for (int i = 0; i < str1.size(); i++) {  if (str1[i] == ' ') {//проверяем есть ли текущее слово в другой строке  FIND = str1.substr(i - count\_s, count\_s);//с помощью subtract извлекаем текущее слово  pos = str1.find(FIND);//смотрим есть ли оно в строке 2  if (pos < i - count\_s) {//если выводится что-то больше -1,то значит слово есть    str1.erase(i - count\_s, count\_s+1);//удаляем его  i -= count\_s;//возвращаемся на то место в строке, где находились до обработки нечетного слова  count\_s = 1;  }  else { count\_s = 0; }//чтобы не считывать пробел между словами как часть слова  }  else{ count\_s++;}//считаем кол-во символов в слове    }  cout << str1;//выводим обновленную строку p  return 0;  } |
| **Результат программы** |
|  |

5. Даны два предложения. Найти самое короткое из слов первого предложения, которого нет во втором предложении.

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <string>//данная директива нужна для использования типа данных string  using namespace std;  int main() {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  string str1, str2, FIND;//объявляем строки  int count\_s = 0, pos,ind=0,min\_word;  cout << "Введите строку S1:"; getline(cin, str1);//при помощи функции getline мы можем считать строку вместе с пробелами  cout << "Введите строку S2:"; getline(cin, str2);  min\_word = str1.size();  str1.insert(str1.size(), " ");//добавляем в конце пробел для возможности обработки последнего  str2.insert(str2.size(), " ");//добавляем в конце пробел для возможности обработки последнего слова  for (int i = 0; i < str1.size(); i++) {  if (str1[i] == ' ') {//проверяем есть ли текущее слово в другой строке  FIND = str1.substr(i - count\_s, count\_s);//с помощью subtract извлекаем текущее слово  pos = str2.find(FIND);//смотрим есть ли оно в строке 2  if (pos == 0) {//если слово первое во второй строке,то ищем пробел только после самого слова  if ( str2[pos+count\_s] == ' ') { count\_s = 0; }//проверяем явлется ли текущее слово отдельным словом,а не частью другого  else {  if (min\_word > count\_s) {//смотрим является ли это слово наименьшим и если так,то записываем его размер и индекс  min\_word = count\_s;  ind = i - count\_s;  };  count\_s = 0;  }  }    if (pos > 0) {//ищем пробелы до и после слова  if(str2[pos - 1] == ' ' and str2[pos + count\_s] == ' '){ count\_s = 0; }  else {  if (min\_word > count\_s) {//смотрим является ли это слово наименьшим и если так,то записываем его размер и индекс  min\_word = count\_s;  ind = i - count\_s;  };  count\_s = 0;  }  }  if (pos < 0) {  if (min\_word > count\_s) {//смотрим является ли это слово наименьшим и если так,то записываем его размер и индекс  min\_word = count\_s;  ind = i - count\_s;  };  count\_s = 0;  }    }  else { count\_s++; }//считаем кол-во символов в слове  }  cout << str1.substr(ind,min\_word);//выводим наименьшее слово  return 0;  } |
| **Результат программы** |
|  |