Беларускі дзяржаўны тэхналагічны ўніверсітэт

Факультэт інфармацыйных тэхналогій

Кафедра праграмнай інжэнерыі

Лабараторная работа 13

Па дысцыпліне «Асновы алгарытмізацыі і праграмавання»

На тэму «Апрацоўка сімвальнай інфармацыі»

Выканаў:

Студэнт 1 курса 6 группы

Гармаза Максім Сяргеевіч

Выкладчык: асс. Андронава М.В.

2023, Мінск

Варыянт 3

Заданне 6

|  |
| --- |
| 1. В строке есть два символа \*. Получить все символы между первым и вторым символом \*. |
| #include <iostream>  #include <windows.h>  using namespace std;  // Функцыя для вываду радка ў кансоль  // \*радка?  void printString(int start, int end, char str[]) {  for (int pos = start + 1; pos < end; pos++) {  cout << str[pos];  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  char str[200];  int start = 0, end = 0;  cout << "Увядзіце радок: "; cin >> str;  //Тут мы глядзім кожны сімвал радка і калі гэта \*, то:  for (int i = 0; i < str[i] != 0; i++) {  if (str[i] == '\*' && start == end) { //Паколькі дакладна з умовы можна сцвярджаць, што тут больш за 1 \*, то індэксы першага і другога будуць адрознівацца. таму мы глядзім ці аднолькавыя яны ў зменных. калі так, то гэта значыць, што гэта першы сімвал \*. інкаш гэта другі сімвал  start = i;  }  else if (str[i] == '\*') {  end = i;  }  }  printString(start, end, str);  return 0;  } |
|  |
| 2. Дана строка, состоящая из букв, цифр, запятых, точек, знаков + и –. Выделить подстроку, которая соответствует записи целого числа. |
| #include <windows.h>  #include <iostream>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, char\* ptr) {  for (int pos = 0; pos < size; pos++) {  cout << \*(ptr + pos);  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  char str[200], numbers[100];  char\* ptr = str, \* ptrn = numbers;  int counter = 0;  cout << "Увядзіце радок: "; cin.getline(str, sizeof(str)); //выкарыстоўваем функцыю для ўвода радка каб пазбегнуть праблем з некаторымі сімваламі  //Тут мы карыстаемся паказальнікамі ў памяці  for (int i = 0; \*(ptr + i) != 0; i++) { // \*(ptr + i) != 0 будзе значыць, что мы дайшлі для сімвала \0 што з'яўляецца сімвалам канца стракі і далей ісці не трэба  if ('0' <= \*(ptr + i) && \*(ptr + i) <= '9') { //параўноўваем сімвалы радка з сімваламі 0 і 9 і калі ўмова выконваецца, то гэта лічба  \*(ptrn + counter++) = \*(ptr + i); //захоўваем лічбу ў новы масіў  }  }  //выводзім масіў карыстаючыся функцыяй  printArray(counter, ptrn);  return 0;  } |
|  |

Варыянт 4

|  |
| --- |
| 1. Написать программу, которая удаляет в строке все буквы **b** в тексте, написанном латинскими буквами. |
| #include <windows.h>  #include <iostream>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, char arr[]) {  for (int pos = 0; pos < size; pos++) {  cout << arr[pos];  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  char str[200], b\_deteled[100];  char\* ptr = str, \* ptrn = b\_deteled;  int counter = 0;  cout << "Увядзіце радок: "; cin >> str;  //Тут мы карыстаемся паказальнікамі ў памяці  for (int i = 0; str[i] != 0; i++) { // str[i] != 0 будзе значыць, что мы дайшлі для сімвала \0 што з'яўляецца сімвалам канца радка і далей ісці не трэба  if (str[i] != 'b') { //калі гэта не b  b\_deteled[counter++] = str[i]; //захоўваем сімвал ў новы масіў  }  }  //выводзім масіў карыстаючыся функцыяй  printArray(counter, b\_deteled);  return 0;  } |
|  |
| 2. Дана строка символов, состоящая из произвольных десятичных цифр, разделенных пробелами. Найти числа этой строки в порядке возрастания их значений. |
| #include <windows.h>  #include <iostream>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, char\* ptr) {  for (int pos = 0; pos < size; pos++) {  cout << \*(ptr + pos)<< ' ';  }  cout << '\n';  }  void bubbleSort(char\* arr, int size) { //Функцыя сартыроўкі шляхам перастановак. калі, напрыклад, гэты элемент 7, наступны 5, то тады мяняем гэтыя элементы ў масіве месцамі. і так да таго пакуль увесь масіў не будзе  адсартаваны  for (int i = 0; i < size - 1; i++) {  for (int j = 0; j < size - i - 1; j++) {  if (\*(arr+j) > \*(arr+j+1)) {  // Мяняем пазіцыямі элементы калі гэты лік больш наступнага  char temp = \*(arr+j);  \*(arr+j) = \*(arr+j+1);  \*(arr+j+1) = temp;  }  }  }  }  int main()  {  SetConsoleOutputCP(1251);  char str[200], numbers[100];  char\* ptr = str;  char \*ptr\_n = numbers;  int counter = 0;  cout << "Увядзіце радок: "; cin.getline(str, sizeof(str));  //Тут мы карыстаемся паказальнікамі ў памяці  for (int i = 0; i < \*(ptr + i) != 0; i++) { // \*(ptr + i) != 0 будзе значыць, что мы дайшлі для сімвала \0 што з'яўляецца сімвалам канца стракі і далей ісці не трэба  if (\*(ptr + i) != ' ') { //дабаўляем усе лікі і ігнаруем прабел  \*(ptr\_n + counter++) = \*(ptr + i); //захоўваем лічбу ў новы масіў перад гэтым cканвертавав да тыпу int  }  }  //выводзім неадсартаваны масіў карыстаючыся функцыяй  printArray(counter, ptr\_n);  //сартыруем масіў  bubbleSort(ptr\_n, counter);  //выводзім адсартаваны масіў  printArray(counter, ptr\_n);  return 0;  } |
|  |

Варыянт 6

|  |
| --- |
| 1. В строке есть символы **\***. Преобразовать строку следующим образом: удалить все символы **\***, и повторить каждый символ, отличный от **\***. |
| #include <windows.h>  #include <iostream>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, char str[]) {  for (int pos = 0; pos < size; pos++) {  cout << str[pos];  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  char str[200], new\_str[100];  int counter = 0;  cout << "Увядзіце радок: "; cin.getline(str, sizeof(str));  //Тут мы карыстаемся паказальнікамі ў памяці  for (int i = 0; i < str[i] != 0; i++) { // \*(ptr + i) != 0 будзе значыць, что мы дайшлі для сімвала \0 што з'яўляецца сімвалам канца стракі і далей ісці не трэба  if (str[i] != '\*') { //дабаўляем усе сімвалы акрамя \*  new\_str[counter++] = str[i];  new\_str[counter++] = str[i];  }  }  printArray(counter, new\_str);  return 0;  } |
|  |
| 2. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Найти порядковый номер слова минимальной длины и количество символов в этом слове. |
| #include <windows.h>  #include <iostream>  using namespace std;  int main()  {  SetConsoleOutputCP(1251);  char str[200];  char\* ptr = str;  int counter = 0, result = 0, index = 1, counter\_of\_spaces = 1;  cout << "Увядзіце радок: "; cin.getline(str, sizeof(str));    for (int i = 0; \*(ptr + i) != 0; i++) { // \*(ptr + i) != 0 будзе значыць, что мы дайшлі для сімвала \0 што з'яўляецца сімвалам канца радка і далей ісці не трэба  if (\*(ptr + i + 1) == 0) { //для адзінага выпадка, калі даходзіць да канца радка(\0), то гэта не лічыцца за прабел і таму код для апошняга слова не быў бы выкананы. там карыстаюсь гэтай праверкай і павялічваю прэфіксным запісам значэнне result паколькі апошні сімвал слова таксама не улічваецца  if (++counter < result) {  result = counter;  index = counter\_of\_spaces;  }  }  else if (\*(str + i) == ' ') { //калі гэта прабел, то мы глядзім: калі result больш за лічыльнік, то мяняем яго значэнне на знач. лічыльніка і захоўваем яго нумар. ў любым выпадку абнуляем лічыльнік  if (counter < result) {  index = counter\_of\_spaces;  result = counter;  }  result = (result == 0) ? counter : result; //паколькі 0 заўседы буддзе меншым, правяраем как result быў нероўны 0, інакш result атрымліваем значэнне для першага элемента  counter = 0;  counter\_of\_spaces++;  }  else { //калі гэта не прабел, то павялічваем лічыльнік  counter++;  }  }  cout <<"Мінімальн. даўж. слова: " << " " << result<<endl;  cout << "Нумар гэтага слова: " << index;  return 0;  } |
|  |

Варыянт 11

|  |
| --- |
| 1. Вывести текст, составленный из последних букв всех слов. |
| #include <windows.h>  #include <iostream>  using namespace std;  void printArray(int size, char str[]) {  for (int pos = 0; pos < size; pos++) {  cout << str[pos];  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  char str[200], result[200];  int counter = 0;  cout << "Увядзіце радок: "; cin.getline(str, sizeof(str));  for (int i = 0; str[i] != 0; i++) {  if (str[i] == ' ') { //Калі слова скончылася(калі гэта прабел)  result[counter++] = str[i - 1]; //дабаўляем папярэдні сімвал у выніковы масіў  }  else if (str[i + 1] == 0) { //паклькі радок заканчваецца не прабелам, то даўляем гэтую праверку для ўліку паследняга слова  result[counter++] = str[i];  }  }  printArray(counter, result);  return 0;  } |
|  |
| 2. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Найти порядковый номер слова максимальной длины и номер позиции в строке, с которой оно начинается. |
| #include <windows.h>  #include <iostream>  using namespace std;  int main()  {  SetConsoleOutputCP(1251);  char str[200];  char\* ptr = str;  int counter = 0, result = 0, index = 1, counter\_of\_spaces = 1;  cout << "Увядзіце радок: "; cin.getline(str, sizeof(str));  for (int i = 0; \*(ptr + i) != 0; i++) { // \*(ptr + i) != 0 будзе значыць, что мы дайшлі для сімвала \0 што з'яўляецца сімвалам канца радка і далей ісці не трэба  if (\*(ptr + i + 1) == 0) { //для адзінага выпадка, калі даходзіць да канца радка(\0), то гэта не лічыцца за прабел і таму код для апошняга слова не быў бы выкананы. там карыстаюсь гэтай праверкай і павялічваю прэфіксным запісам значэнне result паколькі апошні сімвал слова таксама не улічваецца  if (++counter > result) {  result = counter;  index = counter\_of\_spaces;  }  }  else if (\*(str + i) == ' ') { //калі гэта прабел, то мы глядзім: калі result больш за лічыльнік, то мяняем яго значэнне на знач. лічыльніка і захоўваем яго нумар. ў любым выпадку абнуляем лічыльнік  if (counter > result) {  index = counter\_of\_spaces;  result = counter;  }  result = (result == 0) ? counter : result; //паколькі 0 заўседы буддзе меншым, правяраем как result быў нероўны 0, інакш result атрымліваем значэнне для першага элемента  counter = 0;  counter\_of\_spaces++;  }  else { //калі гэта не прабел, то павялічваем лічыльнік  counter++;  }  }  cout << "Максімальная даўж. слова: " << " " << result << endl;  cout << "Нумар гэтага слова: " << index;  return 0;  } |
|  |