Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 13

По дисциплине «Основы алгоритмизации и программирования»

На тему «Обработка символьной информации»

Выполнила:

Студент(ка) 1 курса 7 группы

Подшиваленко Диана Игоревна

Преподаватель: асс. Андронова М.В.

2023, Минск

|  |  |
| --- | --- |
| **11** | 1. Вывести текст, составленный из последних букв всех слов.  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 <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i = 0;

char t[200]; // массив для хранения строки

gets\_s(t); // считываем строку

while(t[i] != '\0') {

if (t[i + 1] == ' ' or t[i + 1] == '\0') { // выводим последние символы слов

cout << t[i];

}

i++;

}

}



#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i = 0, pos = 0, len = 0, num = 0, max\_len = 0, pos\_slova = 0, num\_slova = 0;

bool k = false;

char t[200]; // массив для хранения строки

gets\_s(t); // считываем строку

while(\*(t + i) != '\0') {

if (\*(t + i) != ' ' and k == false) { // ищем начало слова

k = true;

pos = i;

len += 1;

num += 1;

}

else if (\*(t + i) != ' ' and k == true and \*(t + i + 1) != '\0') { // если мы находимся в слове, то прибавляем +1 к длине

len += 1;

}

else if (\*(t + i) == ' ' or \*(t + i + 1) == '\0') { // если конец слова

if (\*(t + i + 1) == '\0') {

len += 1;

}

k = false;

if (len > max\_len) { // проверяем длину максимального слова и нового слова

max\_len = len;

pos\_slova = pos;

num\_slova = num;

}

len = 0;

}

i++;

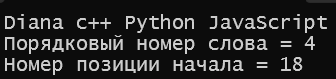
}

cout << "Порядковый номер слова = " << num\_slova << endl;

cout << "Номер позиции начала = " << pos\_slova + 1 << endl;

}

// я начинала счёт позиций с номера 1



**Дополнительные задания**

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

#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char t[200]; // массив для хранения строки

gets\_s(t); // считываем строку

bool k = false;

char a;

int pos = 0;

cout << "Введите окончание слова (1 символ) " << endl;

cin >> a;

int i = 0;

while (t[i] != '\0') { // пока не конец строки

if (t[i] != ' ' and k == false) { // ищем начало слова и фиксируем позицию

k = true;

pos = i;

}

else if (t[i + 1] == ' ' and t[i] != a) { // если слово не заканчивается на символ, то просто указываем, что слово закончилось

k = false;

}

else if ((t[i + 1] == ' ' and t[i] == a and k == true) or (t[i + 1] == '\0' and t[i] == a and k == true)) { // если слово заканчивается на символ, то выводим его

for (int j = pos; j <= i; j++) {

cout << t[j];

}

cout << endl;

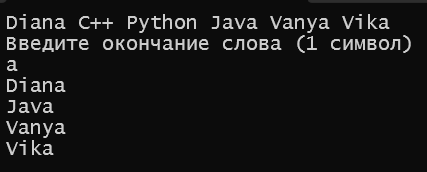
k = false;

}

i++;

}

}



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

#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

#include <vector>

#include <string>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

vector <string> a; // я использовала вектор, потому что мне так проще, надеюсь, что Вы не против

char s[500];

gets\_s(s);

int i = 0;

bool t = false;

string k = ""; // создаём строку для наших слов

while (s[i] != '\0') { // пока не конец строки

if (s[i] != ' ' and t == false) { // фиксируем начало слова

t = true;

k += s[i];

}

else if (s[i] != ' ' and t == true and s[i + 1] != '\0') { // проходим по слову и добавляем элементы в строку k

k += s[i];

}

else if ((s[i] == ' ' and t == true) or (s[i + 1] == '\0' and t == true)) { // если конец слова, то

if (s[i + 1] == '\0') {

k += s[i];

}

t = false; // фиксируем конец слова

if (find(begin(a), end(a), k) == end(a)) { // ищем его в векторе, если нет, то добавляем в вектор

a.push\_back(k);

}

k = ""; // очищаем нашу строку для слов

}

i++;

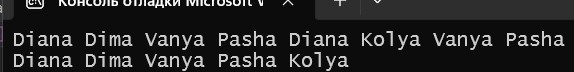
}

for (int i = 0; i < a.size(); i++) { // выводим наши слова

cout << a[i] << ' ';

}

}



**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.**

#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

#include <vector>

#include <string>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

vector <string> a; // я использовала вектор, потому что мне так проще, надеюсь, что Вы не против

char s[500];

gets\_s(s);

int i = 0, num = 1; //переменная num для определения порядка слова

string k = ""; // создаём строку для наших слов

while (s[i] != '\0') { // пока не конец строки

if (num % 2 == 0) { // если слово стоит на чётной позиции

while (s[i] != ' ' and s[i] != '\0') { // проходим по элементам и прибавляем их к строке k, пока не наткнёмся на пробел или конец строки

k = s[i] + k;

i++;

}

a.push\_back(k); // добавляем слово в наш вектор слов

k = ""; // очищаем строку k

num += 1;

i++;

}

else if (num % 2 != 0) { // если число стоит на нечётной позиции

while (s[i] != ' ' and s[i + 1] != '\0') { // проходим по элементам, пока не наткнёмся на пробел или конец строки

i++;

}

num += 1;

i++;

}

}

for (int i = 0; i < a.size(); i++) { // выводим наши слова

cout << a[i] << ' ';

}

}

