Беларускі дзяржаўны тэхналагічны ўніверсітэт

Факультэт інфармацыйных тэхналогій

Кафедра праграмнай інжэнерыі

 Лабараторная работа 13

Па дысцыпліне «Асновы алгарытмізацыі і праграмавання»

На тэму « Апрацоўка сімвальнай інфармацыі»

Выканала:

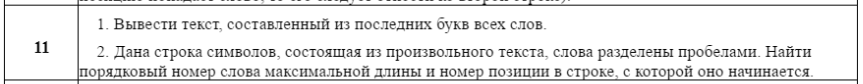
Студэнтка 1 курса 6 группы

Лускіна Вольга Аляксандраўна

Выкладчык: асс. Андронава М.В.

2023, Мінск

Варыянт 11



Заданне 1:

#include <iostream>

#include <Windows.h>

using namespace std;

void main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i = 0;

char a[333];

cout << "Увядзіце пачатковы тэкст: ";

gets\_s(a); // Чытаем радок

cout << "Тэкст, які ў нас атрымаўся: ";

while (a[i] != '\0') { // Глядзім, наступны сімвал гжта прабел ці канец радка

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

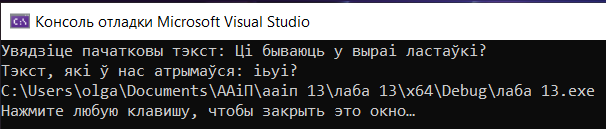
cout << a[i]; // Вывад бягучага сімвала

}

i++; // Павялічваем індэкс, каб перайсці да наступнага сімвала

}

}



Заданне 2:

#include <iostream>

#include <Windows.h>

using namespace std;

void main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i = 0, pach = 0, dauzh = 0, num = 0, max\_dauzh = 0, pach\_slova = 0, num\_slova = 0; // Для пачатка бягучага слова; падліка даўжыні бягучага слова; падліка нумара бягучага слова; захоўвання даўжыні самага доўгага слова, захоўвання яго пазіцыі, нумара

bool k = false;

char a[333];

cout << "Увядзіце радок: ";

gets\_s(a); // Чытаем радок

while (\*(a + i) != '\0') {

if (\*(a + i) != ' ' and k == false)// Шукаем пачатак слова

{

k = true;

pach = i;

dauzh += 1;

num += 1;

} // Калі мы знаходзімся ў слове, то дадаем 1 да даўжыні

else if (\*(a + i) != ' ' and k == true and \*(a + i + 1) != '\0')

{

dauzh += 1;

}

else if (\*(a + i) == ' ' or \*(a + i + 1) == '\0') // Калі канец слова

{

if (\*(a + i + 1) == '\0')

{

dauzh += 1;

} // Параўноўваем даўжыню найдаўжэйшага слова

k = false;

if (dauzh > max\_dauzh)

{

max\_dauzh = dauzh;

pach\_slova = pach;

num\_slova = num;

}

dauzh = 0;

}

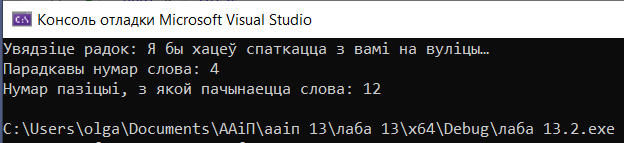
i++;

}

cout << "Парадкавы нумар слова: " << num\_slova << endl;

cout << "Ноумар пазіцыі пачатка: " << pach\_slova + 1 << endl; // Каб падлік пачынаўся не з 0, а з 1

}



Дадатковыя заданни:

Заданне 1:



#include <iostream>

#include <string> // Для працы з радкамі

#include <algorithm> // Для працы з алгарытмамі

#include <vector> // Для працы з вектарамі

#include <Windows.h>

using namespace std;

bool Anagrami(const string& Slova1, const string& Slova2)

{

if (Slova1.size() != Slova2.size()) // Глядзім, ці маюць словы аднолькавую даўжыню

return false;

string adsartSlova1 = Slova1;

string adsartSlova2 = Slova2;

sort(adsartSlova1.begin(), adsartSlova1.end()); // Сартыруем сімвалы ў першым слове

sort(adsartSlova2.begin(), adsartSlova2.end()); // Сартыруем сімвалы ў другім слове

return adsartSlova1 == adsartSlova2; //

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

vector<string> Slounik; // Вектар для захавання слоў

int amount;

string Slova;

cout << "Увядзіце, колькі слоў будзе ў слоўніку: ";

cin >> amount;

cout << "Увядзіце гэтыя словы: ";

for (int i = 0; i < amount; i++)

{

cin >> Slova;

Slounik.push\_back(Slova); // Дадаем слова ў канец вектара

}

cout << "Групы анаграм:" << endl;

vector<bool> visited(Slounik.size(), false); // Вектар, каб сачыць за прагледжанымі словамі

for (int i = 0; i < Slounik.size(); i++)

{

if (visited[i]) // Калі слова ўжо прагледжана, пераходзім да наступнай ітэрацыі

continue;

vector<string> group;

group.push\_back(Slounik[i]); // Дадаем бягучае слова да групы анаграм

for (int j = i + 1; j < Slounik.size(); j++)

{

if (!visited[j] && Anagrami(Slounik[i], Slounik[j])) // Калі словы з'яўляецца анаграмамі і слова не прагледжана

{

group.push\_back(Slounik[j]); // Дадаем слова ў групу анаграм

visited[j] = true; // Адзначаем слова як прагледжанае

}

}

if (group.size() > 1) // Калі ў групе больш за адно слова, выводзім групу

{

cout << "Група: " ;

for (const string& Slova : group)

{

cout << Slova << ' ';

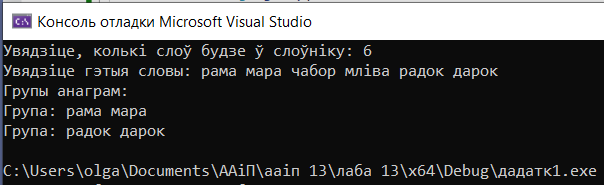
}

cout << endl;

}

}

}



Заданне 2:



#include <iostream>

#include <Windows.h>

#include <vector>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

vector <string> n; // Вектар для захоўвання ўнікальных слоў

int i = 0;

bool t = false;

char a[333];

cout << "Увядзіце сказ: ";

gets\_s(a); // Чытаем радок

cout << "Новы сказ: ";

string r; // Зменная для фарміравання слова

while (a[i] != '\0') { // Пакуль не канец радка

if (a[i] != ' ' and t == false) // Калі бягучы сімвал не прабел і не стаіць сцяг пачатка слова

{

t = true;

r += a[i]; // Дадаем сімвал да бягучага слова

}

else if (a[i] != ' ' and t == true and a[i + 1] != '\0') { // Калі бягучы сімвал не прабел, стаіць сцяг пачатка слова і наступны сімвал не з'яўляецца канцом радка

r += a[i]; // Дадаем сімвал да бягучага слова

}

else if ((a[i] == ' ' and t == true) or (a[i + 1] == '\0' and t == true)) // Калі бягучы сімвал прабел і стаіць сцяг пачатка слова або наступны сімвал з'яўляецца канцом радка і стаіць сцяг пачатка слова

{

if (a[i + 1] == '\0') { // Калі наступны сімвал - канец радка

r += a[i]; // Дадаем сімвал да бягучага слова

}

t = false;

if (find(begin(n), end(n), r) == end(n)) { // Глядзім, ці ёсць бягучае слова ў вектары

n.push\_back(r); // Калі слова няма, дадаем яго ў вектар

}

r = ""; // Чысцім зменную для фарміравання слова

}

i++;

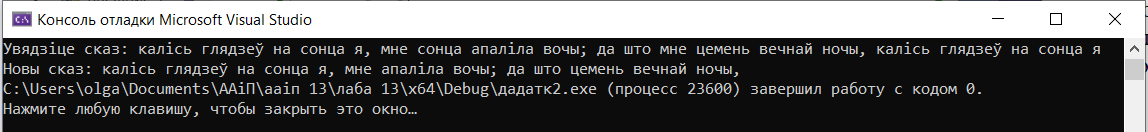
}

for (int i = 0; i < n.size(); i++) { // Выводзім унікальныя словы

cout << n[i] << ' ';

}

}



Заданне 3:



#include <iostream>

#include <vector>

#include <algorithm>

#include <string>

#include <Windows.h>

using namespace std;

string peraculenaeSlova(const string& slova) { // Функцыя, з дапамогай якой мы перакулім слова

string peraculenaeSlova = slova;

int length = slova.length(); // Вызначаем даўжыню слова

for (int i = 0; i < length / 2; i++) { // Перабіраем сімвалы да паловы, бо потым будзем мяняць іх месцамі

swap(peraculenaeSlova[i], peraculenaeSlova[length - i - 1]); // Мяняем сімвалы з пачатка слова з сімваламі з канца

}

return peraculenaeSlova;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

string skaz;

cout << "Увядзіце сказ: ";

getline(cin, skaz);

vector<string> slovi;

string slova;

int index = 0;

for (int i = 0; i < skaz.length(); i++) {

if (skaz[i] == ' ') {

if (index % 2 != 0) { // Глядзім, ці з'яўляецца індэкс цотным

slovi.push\_back(peraculenaeSlova(slova)); // Дадаем перакуленая слова ў вектар

}

slova = ""; // Чысцім зменную для фарміравання слова

index++;

}

else {

slova += skaz[i]; // Дадаем сімвал да бягучага слова

}

}

if (!slova.empty() && index % 2 != 0) {

slovi.push\_back(peraculenaeSlova(slova)); // Дадаем апошняе перакуленае слова ў вектар

}

string result;

for (const string& w : slovi) { // Фарміруем радок з ператворанымі словамі

result += w + " ";

}

cout << "Новы сказ: " << result << endl;

return 0;

}

