4.5 - Labs (10)

* **Lab 4.5.2 (1) Text manipulation: duplicate white space [B]**

#include<cstdlib>

#include<ctime>

using namespace std;

int main()

{

char IP[16];

char\* endptr;

long int a;

cout << "Enter IP: ";

cin >> IP;

if (a = strtol(IP, &endptr, 10))

if (a >= 0 && a <= 255)

{

int i;

for (i = 0; i < 3; i++)

{

++endptr;

if (!(a = strtol(endptr, &endptr, 10)) || !(a >= 0 && a <= 255))

{

cout << "String is not IP";

break;

}

}

if (i < 3)

cout << "IP isn't correct";

else

cout << "Correct IP";

}

else

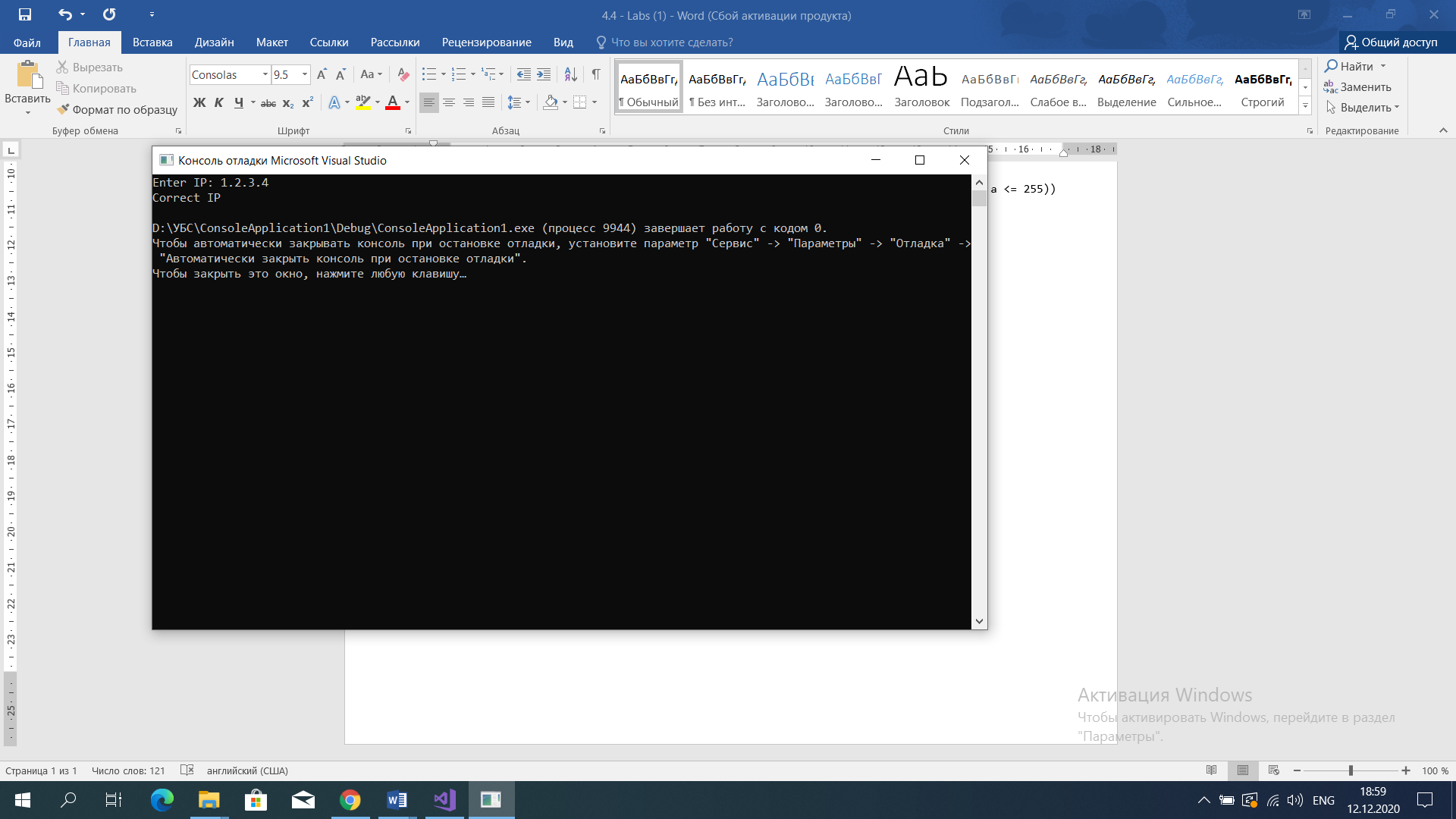
cout << "IP isn't correct";

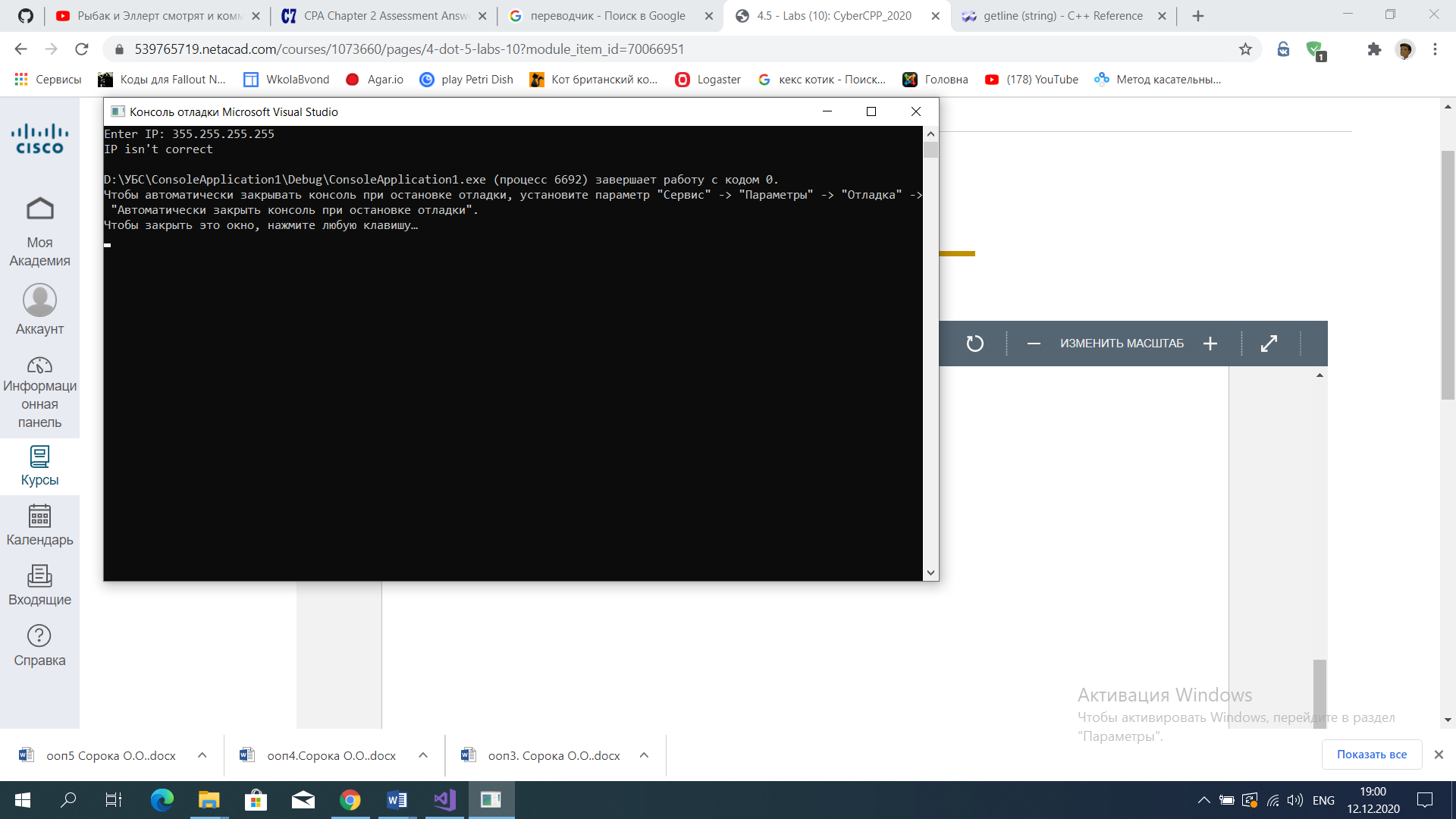
else

cout << "IP isn't only numbers";

cout << endl;

return 0; }





* **Lab 4.5.6 (1) Text manipulation: eliminate duplicates [B]**

#include <iostream>

#include<cstdlib>

#include<ctime>

#include<string>

using namespace std;

int main()

{

string sentence;

cout << "Enter sentence: ";

getline(cin, sentence);

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

if (sentence[i] == ' ' && sentence[i + 1] == ' ')

{

sentence.erase(i, 1);

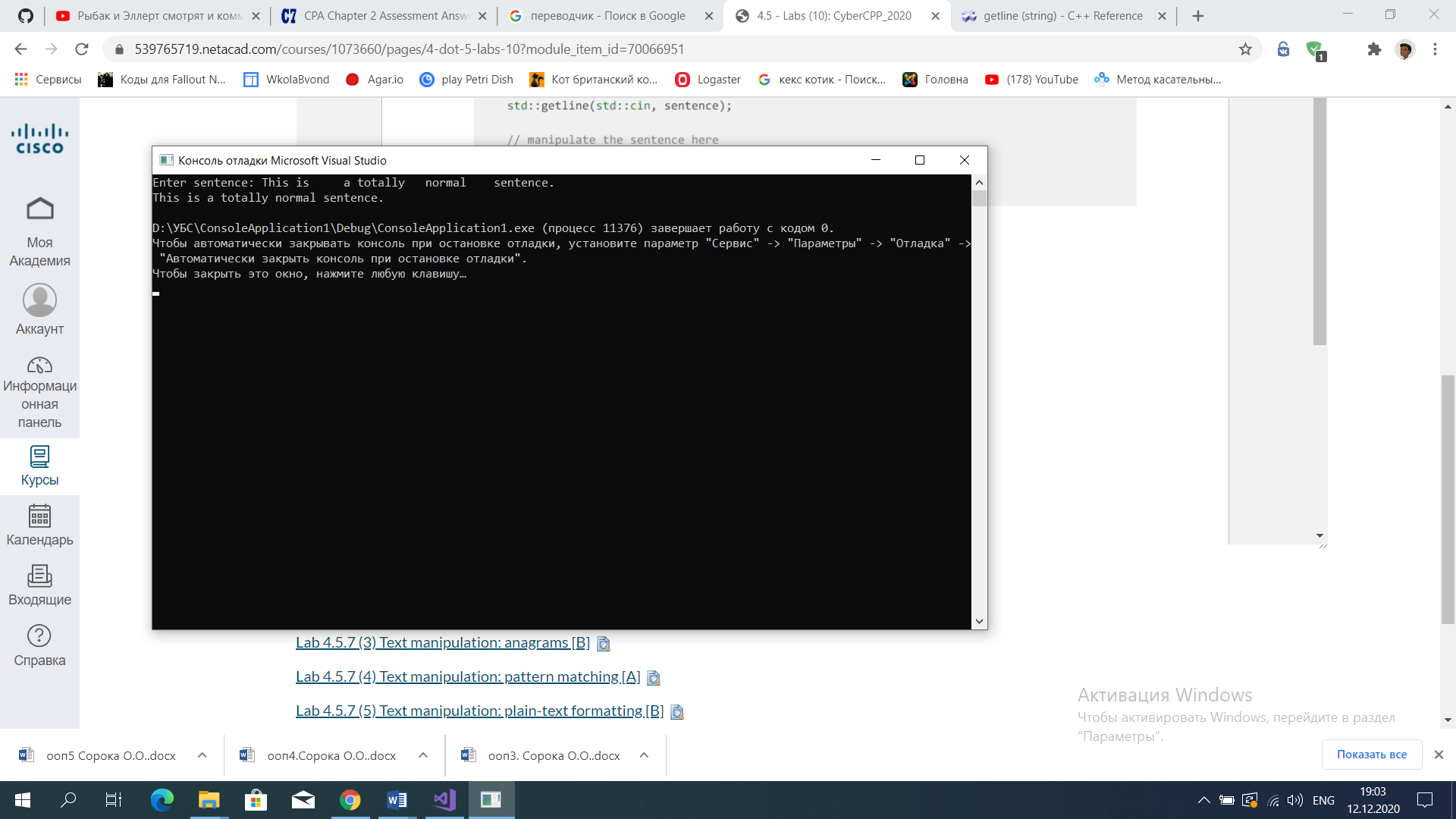
i = (i - 1);

}

cout << sentence << endl;

return 0;

}



* **Lab 4.5.7 (1) Text manipulation: find and replace [A]**

#include <iostream>

#include<cstdlib>

#include<ctime>

#include <vector>

#include <sstream>

using namespace std;

int main()

{

string input;

cout << "Enter the sentence: ";

getline(cin, input);

string temp, output;

vector<string> vec;

stringstream ss;

ss << input;

while (ss >> temp)

vec.push\_back(temp);

int i;

for (i = 0; i < vec.size() - 1; i++)

{

if (vec[i] != vec[i + 1])

output += vec[i] + " ";

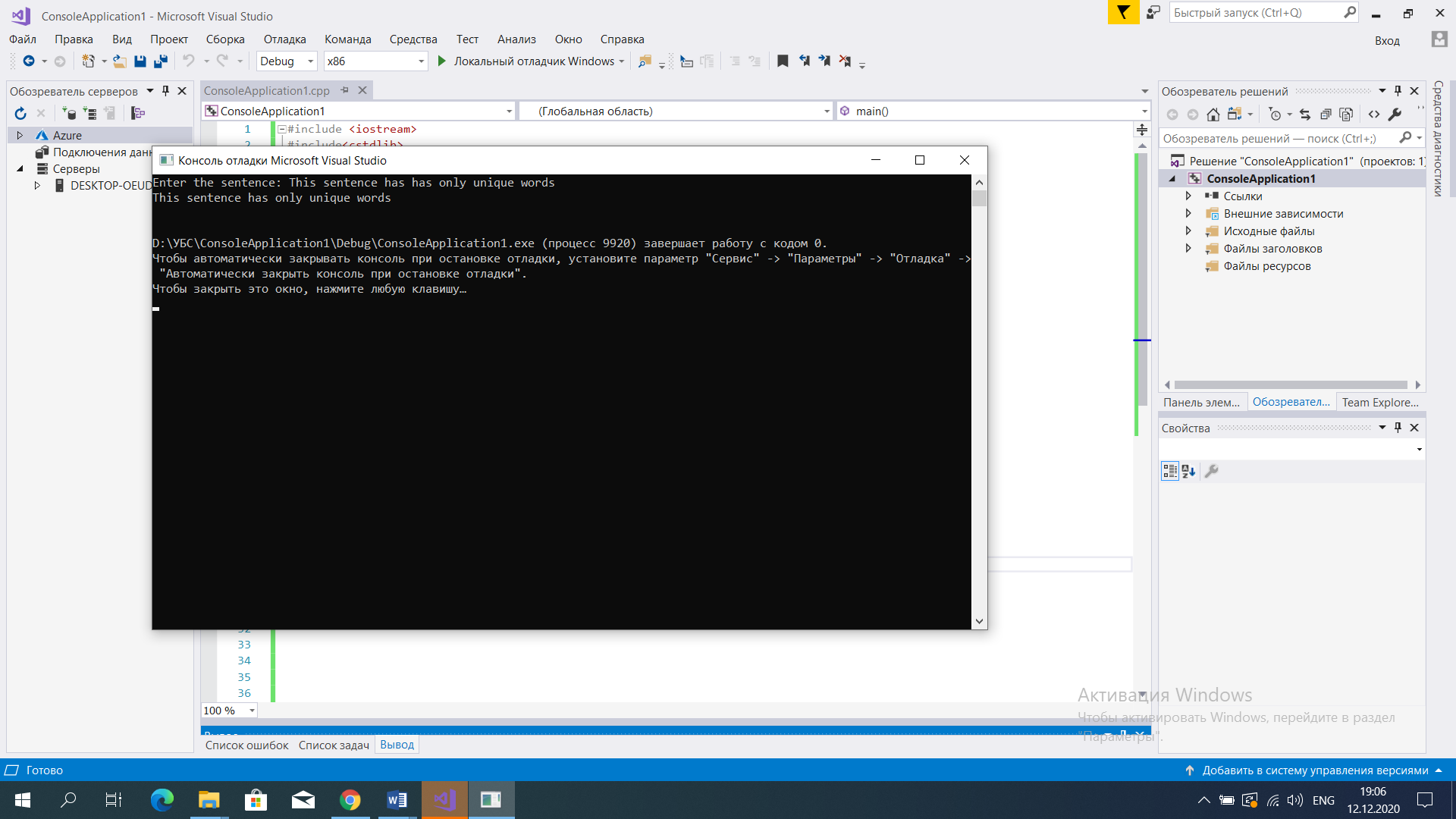
}

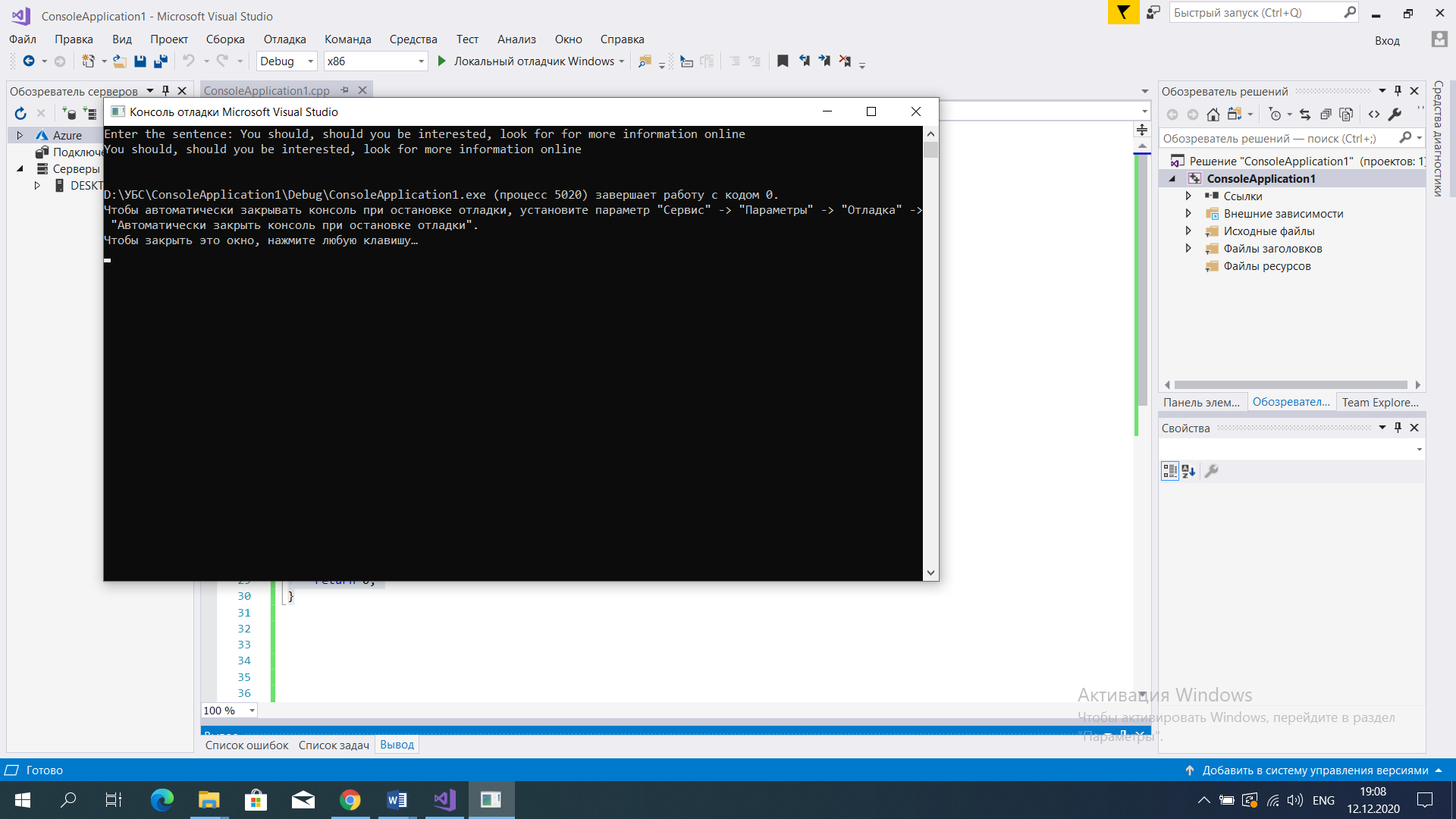
output += vec[i];

cout << output << endl << endl;

return 0;

}





* **Lab 4.5.7 (3) Text manipulation: pattern matching [A]**

#include <iostream>

#include<cstdlib>

#include<string>

#include <algorithm>

using namespace std;

int main()

{

string a, b;

cout << "Enter two words: " << endl;

cin >> a >> b;

if (a.length() == b.length() && a.length() <= 20)

{

sort(a.begin(), a.end());

sort(b.begin(), b.end());

if (a == b)

cout << "Words are anagrams" << endl;

else

cout << "Words aren`t anagrams" << endl;

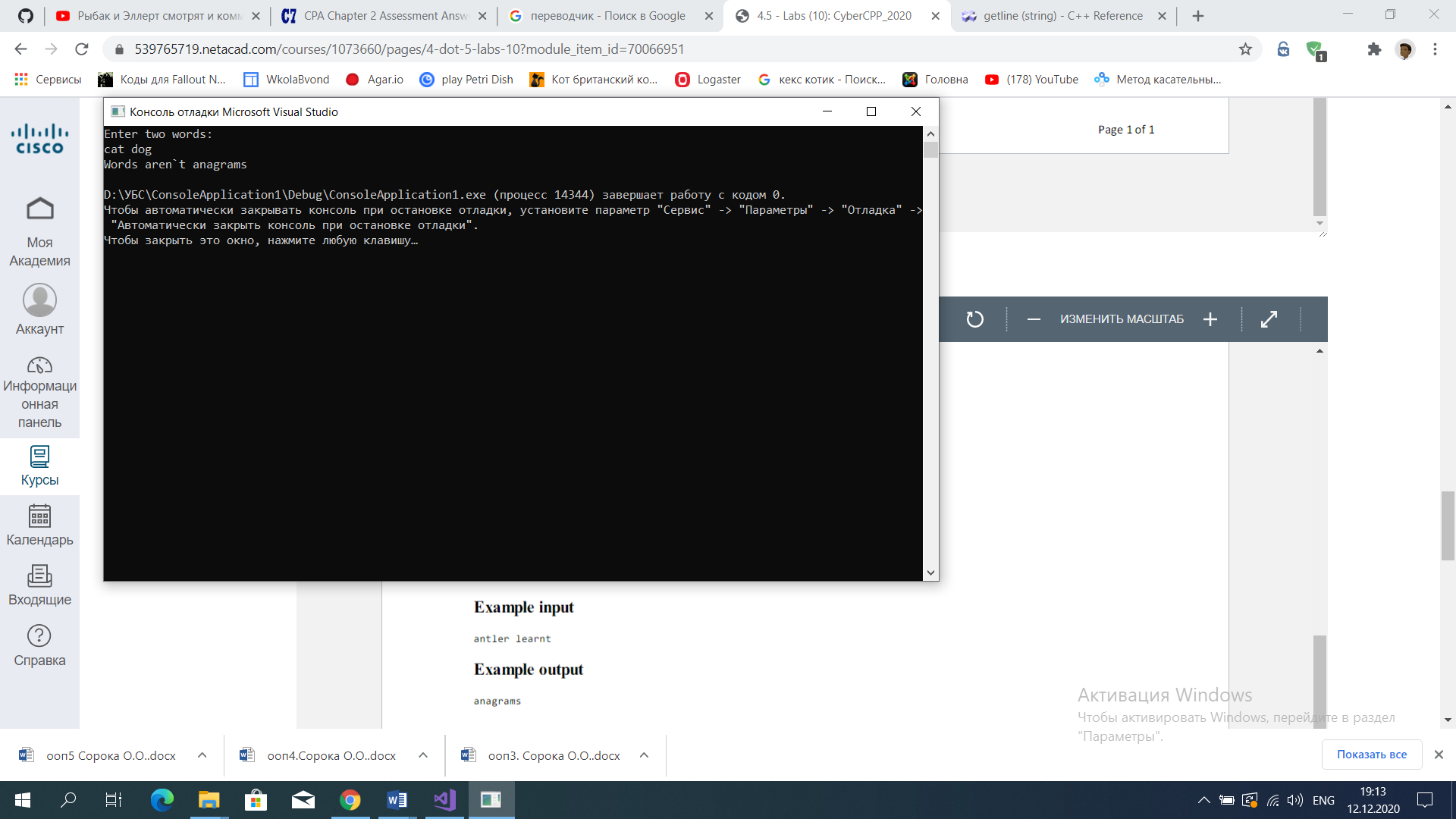
}

else

cout << "Error" << endl;

return 0;

}



* **Lab 4.5.7 (7) Text manipulation: templates [B]**

#include <iostream>

#include<cstdlib>

#include<string>

#include <algorithm>

using namespace std;

int main()

{

string password;

cout << "Enter a password:";

getline(cin, password);

char uc\_letters[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

char letters[] = "abcdefghijklmnopqrstuvwxyz";

char digits[] = "1234567890";

char symbols[] = "/<>,.$%[]{}^&\*!@#\_-+=;:?";

if (password.length() < 8)

cout << "\nThe password must be 8 characters long!\n";

int k = 0;

for (int i = 0; i < 26; i++)

if (password.find(uc\_letters[i]) != string::npos)

k++;

if (k == 0)

cout << "\nThe password must have at least one upper case letter!" << endl;

int x = 0;

for (int i = 0; i < 26; i++)

if (password.find(letters[i]) != string::npos)

x++;

if (x == 0)

cout << "\nThe password must have at least one lower case letter!" << endl;

int y = 0;

for (int i = 0; i < 10; i++)

if (password.find(digits[i]) != string::npos)

y++;

if (y == 0)

cout << "\nThe password must have at least one digit!" << endl;

int z = 0;

for (int i = 0; i < 24; i++)

if (password.find(symbols[i]) != string::npos)

z++;

if (z == 0)

cout << "\nThe password must have at least one special character!" << endl;

if ((password.length()) >= 8 && k != 0 && x != 0 && y != 0 && z != 0)

cout << "\nThe password is valid" << endl;

return 0;

}

