#include<iostream>

using namespace std;

//char\* mystrchr(char\* str, char s) {

// for (size\_t i = 0; str[i]!='\0'; i++)

// {

// if (str[i] == s) {

// return &str[i];

// }

// }

// return nullptr;

//}

//void main() {

// /\*

// 4. char\* mystrchr (char \*str, char s); - funksiya s

//simvolunu str sətrində axtarır əgər taparsa həmin

//hərfə pointer qaytarır, tapmazsa 0.

// \*/

//

// char\* text = new char[100]{};

// cout << "Enter text : ";

// cin.getline(text, 100);

//

// char letter = ' ';

// cout << "Enter letter : ";

// cin >> letter;

// auto result = mystrchr(text,letter);

// if (result != nullptr) {

// cout << result << endl;

// }

// else {

// cout << letter << " does not exist in this text" << endl;

// }

//

//

//

//}

//char\* replace(const char\* text, char letter) {

// int l = strlen(text);

// auto newtext = new char[l + 1]{};

// for (size\_t i = 0; i < l; i++)

// {

// if (text[i] == ' ') {

// newtext[i] = letter;

// }

// else {

// newtext[i] = text[i];

// }

// }

// newtext[l] = '\0';

// return newtext;

//}

//

//

//void main() {

//

// /\*text="salam necesen , hello"

// replace(text,"\*")\*/

//

// char\* text = new char[100]{};

// cin.getline(text, 100);

// auto result = replace(text, ' ');

// cout << result << endl;

//

//

//

// }

//task text de gonderilen elementleri silen funksiya yazin

//

//int getDeletedElementCount(const char\* text, const char& letter) {

// int counter = 0;

// for (size\_t i = 0; i < strlen(text); i++)

// {

// if (text[i] == letter)

// ++counter;

// }

// return counter;

//}

//

//char\* clearText(const char\* text, const char& letter) {

// int count = getDeletedElementCount(text, letter);

// int l = strlen(text) - count;

// auto newtext = new char[l + 1];

// int index = 0;

// for (size\_t i = 0; i < strlen(text); i++)

// {

// if (text[i] != letter) {

// newtext[index] = text[i];

// ++index;

// }

// }

// newtext[l] = '\0';

// return newtext;

//}

//

//

//void main() {

//

// char\* text = new char[100]{};

// cin.getline(text, 100);

// char letter = ' ';

// cin >> letter;

// auto result = clearText(text, letter);

// cout << result << endl;

//

//}

//

//

//char\* getText(const char\* text) {

// char\* newtext;

// for (size\_t i = 0; i < strlen(text)-2; i++)

// {

// if (text[i] == 'n' && text[i + 1] == 'o' && text[i + 2] == 't') {

// newtext = new char[] {"The lyrics is good"};

// return newtext;

// }

// }

// newtext = new char[] {"The lyrics is not good"};

// return newtext;

//}

// //The lyrics is not that poor

//void main() {

// char\* text = new char[100]{};

// cout << "Enter text : ";

// cin.getline(text, 100);

// cout << getText(text) << endl;

//}

/// <summary>

///

/// </summary>

/// <param name="content"> content </param>

/// <param name="tag"> h1 => <h1>content</h1> </param>

/// <returns>html code</returns>

//char\* getHtmlCode(const char\* content, const char\* tag) {

// int l = 2 \* strlen(tag) + 5 + strlen(content);

// auto html = new char[l + 1]{};

// int tag\_l = strlen(tag);

// html[0] = '<';

// html[tag\_l+1] = '>';

//

// for (size\_t i = 0; i < tag\_l; i++)

// {

// html[i + 1] = tag[i];

// }

// int content\_l = strlen(content);

// for (size\_t i = tag\_l+2,i2=0; i2 < content\_l; i++,i2++)

// {

// html[i] = content[i2];

// }

//

// int n\_index = tag\_l + 2 + content\_l;

// html[n\_index] = '<';

// html[n\_index+1] = '/';

// size\_t i = n\_index + 2;

// for (size\_t i2=0; i2 < tag\_l; i++,i2++)

// {

// html[i] = tag[i2];

// }

// html[i] = '>';

// html[i+1] = '\0';

// return html;

//}

//

//void main() {

//

// char\* content = new char[100]{};

// cout << "Enter content : ";

// cin.getline(content, 100);

//

// char\* tag = new char[100]{};

// cout << "Enter tag : ";

// cin.getline(tag, 100);

//

// cout << getHtmlCode(content,tag)<<endl;

//

//}

//93. Write a Python program to extract numbers from a given string.

// Go to the editor

//Sample Output :

//Original string : red 12 black 45 green

//Extract numbers from the said string : [12, 45]