**// In chuỗi ra từng dòng có số thứ tự**

#include <iostream>

#include <string>

using namespace std;

#pragma warning(disable : 4996)

int main()

{

char s1[50];

char s2[] = " ";

cout << "Nhap chuoi: ";

cin.getline(s1, 50);

char \*ptok = strtok(s1, s2);

int i = 0;

while (ptok != NULL)

{

i++;

cout << i <<". "<< ptok << endl;

ptok = strtok(NULL, s2);

}

return 0;

}

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

void Thongke(string s);

void InTuMoiDong(string s);

string ChuanHoa(string s);

bool kiemTraKiTuTrongChuoi(const string& s, char x);

int Sotu(string s);

string Loaibokhoangtrang(string s);

const int SIZE = 256;

#pragma warning(disable : 4996)

int main()

{

string s;

cout << "Nhap vao chuoi: ";

getline(cin, s);

Thongke(s);

InTuMoiDong(s);

cout << "Chuoi sau khi chuan hoa: " << ChuanHoa(s) << endl;

char x;

cout << "Nhap vao ki tu can tim: ";

cin >> x;

if (kiemTraKiTuTrongChuoi(s, x))

{

cout << "Ki tu co trong chuoi" << endl;

}

else {

cout << "Ki tu ko co trong chuoi" << endl;

}

cout << "So tu trong chuoi s: " << Sotu(s) << endl;

cout << "Chuoi sau khi loai bo khoang trang thua: " << Loaibokhoangtrang(s) << endl;

return 0;

}

void Thongke(string s) // Xác định số lần xuất hiện của từng kí tự trong chuỗi.

{

int arrkeys[SIZE] = { 0 };

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

{

if (!isalpha(s[i]))

arrkeys[s[i]]++;

else

{

if (islower(s[i]))

arrkeys[s[i] - 32]++;

else

arrkeys[s[i]]++;

}

}

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

if (arrkeys[i] > 0 && i != 32)

cout << "Ki tu " << char(i) << " co " << arrkeys[i] << " lan xuat hien" << endl;

}

//In các từ trong chuỗi, mỗi từ trên một dòng.

void InTuMoiDong(string s)

{

char\* cstr = new char[s.length() + 1];

strcpy(cstr, s.c\_str());

char\* p = strtok(cstr, " ");

while (p != NULL)

{

cout << p << endl;

p = strtok(NULL, " ");

}

delete[]cstr;

}

string ChuanHoa(string s) //Chuẩn hóa chuỗi(không có các khoảng trắng thừa, các từ được bắt đầu bằng kí tự chữ hoa, các kí tự

/\*còn lại là chữ thường)\*/

{

string res = "";

char \*cstr = new char[s.length() + 1];

strcpy(cstr, s.c\_str());//sao chep chuoi s sang chuoi c\_str de su dung ham strtok

char \*p = strtok(cstr, " ");

while (p != NULL)

{

string str(p);

str[0] = toupper(str[0]);//chuyen ki tu dau thanh hoa

for (int i = 1; i < str.length(); i++)

str[i] = tolower(str[i]);// chuyen cac ki tu con lai sang thuong

res = res + str + " ";

p = strtok(NULL, " ");

}

delete[]cstr;

return res;

}

//Kiểm tra một kí tự có nằm trong chuỗi hay không ?

bool kiemTraKiTuTrongChuoi(const string& s, char x)

{

for (char c : s)

{

if (c == x)

{

return true;

}

}

return false;

}

int Sotu(string s) /\* Đếm số từ trong một chuỗi.\*/

{

char \*cstr = new char[s.length() + 1];

strcpy(cstr, s.c\_str()); //sao chep chuoi s sang chuoi c\_str de su dung ham strtok

char \*p = strtok(cstr, " ");

int dem = 0;

while (p != NULL)

{

dem++;

p = strtok(NULL, " ");

}

delete[]cstr;

return dem;

}

// Loại bỏ các khoảng trắng thừa trong chuỗi.

string Loaibokhoangtrang(string s)

{

string res = "";

char \*cstr = new char[s.length() + 1];

strcpy(cstr, s.c\_str());//sao chep chuoi s sang chuoi c\_str de su dung ham strtok

char \*p = strtok(cstr, " ");

while (p != NULL)

{

string str(p);

res = res + str + " ";

p = strtok(NULL, " ");

}

delete[]cstr;

return res;

}

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

string Tachtudau(string s);

string Tachtucuoi(string s);

string Chuyentucuoithanhdau(string s);

#pragma warning(disable : 4996)

int main()

{

string s;

cout << "Nhap chuoi: ";

getline(cin, s);

cout << "Tu dau: ";

Tachtudau(s);

cout << "Tu cuoi: " << Tachtucuoi(s) << endl;

cout << "Chuyen tu cuoi thanh dau: " << Chuyentucuoithanhdau(s) << endl;

return 0;

}

//Tách từ đầu tiên trong chuỗi.

string Tachtudau(string s)

{

char\* cstr = new char[s.length() + 1];

strcpy(cstr, s.c\_str());

char\* p = strtok(cstr, " ");

cout << p << endl;

p = strtok(NULL, " ");

delete[]cstr;

return p;

}

//Tách từ cuối cùng trong chuỗi

string Tachtucuoi(string s)

{

string x;

char\* cstr = new char[s.length() + 1];

strcpy(cstr, s.c\_str());

char\* p = strtok(cstr, " ");

while (p != NULL)

{

x = p;

p = strtok(NULL, " ");

}

delete[]cstr;

return x;

}

//Chuyển từ cuối cùng thành từ đầu tiên trong chuỗi

string Chuyentucuoithanhdau(string s)

{

string x;

char\* cstr = new char[s.length() + 1];

strcpy(cstr, s.c\_str());

char\* p = strtok(cstr, " ");

while (p != NULL)

{

x = p;

p = strtok(NULL, " ");

}

x = x + " ";

s.insert(0, x);

delete[]cstr;

return s;

}

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

void Intudaingannhat(string s);

string Daochuoi(string s);

bool ChuoiDoiXung(string s);

bool ChuoiTuanHoan(const string& s);

#pragma warning(disable : 4996)

int main()

{

string s;

cout << "Nhap chuoi: ";

getline(cin, s);

Intudaingannhat(s);

cout << "Chuoi sau khi dao: " << Daochuoi(s) << endl;

if (ChuoiDoiXung(s) == true)

cout << "Chuoi doi xung" << endl;

else cout << "Chuoi ko doi xung" << endl;

if (ChuoiTuanHoan(s))

cout << "Chuoi la chuoi tuan hoan" << endl;

else

cout << "Chuoi khong la chuoi tuan hoan" << endl;

return 0;

}

//In từ dài nhất (từ ngắn nhất) trong chuỗi.

void Intudaingannhat(string s)

{

string longestWord;

string shortestWord;

string word;

for (size\_t i = 0; i < s.length(); ++i)

{

if (s[i] != ' ')

{

word += s[i];

}

else

{

if (longestWord.empty() || word.length() > longestWord.length())

{

longestWord = word;

}

if (shortestWord.empty() || word.length() < shortestWord.length())

{

shortestWord = word;

}

word = "";

}

}

if (!word.empty())

{

if (longestWord.empty() || word.length() > longestWord.length())

{

longestWord = word;

}

if (shortestWord.empty() || word.length() < shortestWord.length())

{

shortestWord = word;

}

}

cout << "Tu dai nhat: " << longestWord << endl;

cout << "Tu ngan nhat: " << shortestWord << endl;

}

// Đảo ngược chuỗi.

string Daochuoi(string s)

{

string x;

for (int i = s.length() - 1; i >= 0; --i)

{

x += s[i];

}

return x;

}

//Kiểm tra chuỗi có phải là một palindrome hay không ?

bool ChuoiDoiXung(string s)

{

bool bcheck = true;

int i = 0, j = s.length() - 1;

while (i < j) // tung cap ki tu o hai dau chuoi

{

if (tolower(s[i]) != tolower(s[j]))

{// neu co hai ki tu khac nhau

bcheck = false;

break;

}

}

i++, j++;

return bcheck;

}

// Kiểm tra chuỗi có tuần hoàn hay không ?

bool ChuoiTuanHoan(const string& s)

{

int doDai = s.length();

if (doDai == 0)

{

return false;

}

string ghep = s + s;

for (int i = 1; i < doDai; ++i)

{

string phuChuoi = ghep.substr(i, doDai);

if (phuChuoi == s)

{

return true;

}

}

return false;

}

#include <iostream>

#include <string>

using namespace std;

//string FindAndDel(string s1, string s2);

#pragma warning(disable : 4996)

int main()

{

string s1, s2;

cout << "Nhap chuoi s1: ";

getline(cin, s1);

cout << "Nhap chuoi s2: ";

getline(cin, s2);

//Kiểm tra chuỗi con có nằm trong một chuỗi hay không ?

int pos = s1.find(s2);

if (pos >= 0)

{

cout << "Chuoi con co trong chuoi lon" << endl;

}

else

{

cout << "Chuoi con khong co trong chuoi lon" << endl;

}

/\*cout << "Chuoi sau khi xoa: " << FindAndDel(s1, s2) << endl;\*/

return 0;

}

//Tìm và xóa tất cả

//string FindAndDel(string s1, string s2)

//{

// int pos = s1.find(s2);

// while (pos >= 0)

// {

// s1 = s1.erase(pos, s2.length());

// pos = s1.find(s2);

// }

// return s1;

//}

#include <iostream>

#include <string>

#include <unordered\_map>

using namespace std;

string thayTheKyTu(const string& chuoi, const string& chuoiCon);

int demSoLanXuatHien(const string& chuoi, const string& chuoiCon);

string loaiBoChuoiCon(const string& s, const string& subs);

string chenChuoiCon(const string& s, const string& subs, int k);

string maHoaChuoi(const string& s, const unordered\_map<char, char>& bangThayThe);

string giaiMaChuoi(const string& s, const unordered\_map<char, char>& bangThayThe);

int main()

{

string chuoi;

cout << "Nhap chuoi: ";

getline(cin, chuoi);

string chuoiCon;

cout << "Nhap chuoi con: ";

getline(cin, chuoiCon);

string ketQua = thayTheKyTu(chuoi, chuoiCon);

cout << "Ket qua: " << ketQua << endl;

cout << "So lan xuat hien cua chuoi con: " << demSoLanXuatHien(chuoi, chuoiCon) << endl;

cout << "Chuoi sau khi loai bo chuoi con subs: " << loaiBoChuoiCon(chuoi, chuoiCon) << endl;

int k;

cout << "Nhap vi tri can chen chuoi con: ";

cin >> k;

cout << "Chuoi sau khi chen chuoi con: " << chenChuoiCon(chuoi, chuoiCon, k) << endl;

/\*int main() { \*\*\*Phần mã hoá

string s;

cout << "Nhap chuoi: ";

getline(cin, s);

unordered\_map<char, char> bangThayThe;

bangThayThe['a'] = 'x';

bangThayThe['b'] = 'y';

bangThayThe['c'] = 'z';

string maHoa = maHoaChuoi(s, bangThayThe);

cout << "Chuoi da ma hoa: " << maHoa << endl;

string giaiMa = giaiMaChuoi(maHoa, bangThayThe);

cout << "Chuoi da giai ma: " << giaiMa << endl;

return 0;

}\*/

return 0;

}

//Thay thế các kí tự trong chuỗi nhập bằng các kí tự tương ứng trong chuỗi con.

string thayTheKyTu(const string& chuoi, const string& chuoiCon)

{

string ketQua = chuoi;

int doDaiChuoi = chuoi.length();

int doDaiChuoiCon = chuoiCon.length();

for (int i = 0; i < doDaiChuoi; ++i) {

if (ketQua[i] != ' ') {

ketQua[i] = chuoiCon[i % doDaiChuoiCon];

}

}

return ketQua;

}

//Đếm số lần xuất hiện chuỗi con trong một chuỗi.

int demSoLanXuatHien(const string& chuoi, const string& chuoiCon)

{

int doDaiChuoi = chuoi.length();

int doDaiChuoiCon = chuoiCon.length();

int soLanXuatHien = 0;

for (int i = 0; i <= doDaiChuoi - doDaiChuoiCon; ++i) {

bool timThay = true;

for (int j = 0; j < doDaiChuoiCon; ++j) {

if (chuoi[i + j] != chuoiCon[j]) {

timThay = false;

break;

}

}

if (timThay) {

soLanXuatHien++;

}

}

return soLanXuatHien;

}

// Loại bỏ chuỗi con subs(nếu có) ra khỏi chuỗi s.

string loaiBoChuoiCon(const string& s, const string& subs)

{

string ketQua = s;

size\_t found = ketQua.find(subs);

while (found != string::npos) {

ketQua.erase(found, subs.length());

found = ketQua.find(subs, found);

}

return ketQua;

}

// Chèn chuỗi con subs vào trong chuỗi s tại vị trí k.

string chenChuoiCon(const string& s, const string& subs, int k)

{

string ketQua = s;

ketQua.insert(k, subs);

return ketQua;

}

// Mã hóa và giải mã chuỗi dùng phép thay thế(substitution).

string maHoaChuoi(const string& s, const unordered\_map<char, char>& bangThayThe)

{

string ketQua = s;

for (char& c : ketQua) {

if (bangThayThe.count(c)) {

c = bangThayThe.at(c);

}

}

return ketQua;

}

// Mã hóa và giải mã chuỗi dùng bảng thay thế

string giaiMaChuoi(const string& s, const unordered\_map<char, char>& bangThayThe)

{

string ketQua = s;

for (char& c : ketQua) {

for (const auto& pair : bangThayThe) {

if (pair.second == c) {

c = pair.first;

break;

}

}

}

return ketQua;

}

//Bài 5. Viết chương trình cho phép người dùng nhập password(tối đa 10 ký tự) và hiển thị dưới dạng

//ký tự “\*”.Sau đó kiểm tra xem đúng password là “svCQ” hay không ? Nếu đúng thông báo đã đăng

//nhập thành công, nếu sai yêu cầu nhập lại.Tối đa cho phép nhập 3 lần.

//Ví dụ :

//Nhap mat khau : \*\*\*\*\*\*\*

//Nhap sai!Nhap lai!

//Nhap mat khau : \*\*\*\*

//Dang nhap thanh cong!

#include <iostream>

#include <string>

#include <conio.h>

using namespace std;

const char BACKSPACE = 8; // ma ascii kis tujw backspace

const char ENTER = 13; // ma ascii ki tu enter

string GetPass(string prompt);

int main()

{

int dem = 0;

bool bTrue = true;

string password = "svCQ";

while (true)

{

string input = GetPass("Nhap mat khau: ");

if (input != password)

{

dem++;

if (dem < 3)

cout << "Nhap sai! Nhap lai!" << endl;

else

{

cout << "Khong the dang nhap" << endl;

bTrue = false;

break;

}

}

else break;

}

if (bTrue)

cout << "Dang nhap thanh cong" << endl;

return 0;

}

string GetPass(string prompt)

{

cout << prompt;

string password;

char ch;

while ((ch = \_getch()) != ENTER)

{

if (ch == BACKSPACE)//nhan phim backspace xoa lui

{

if (password.length() > 0)

{

cout << "\b \b";

password.resize(password.length() - 1);//giam 1 ki tu

}

}

else

{

password += ch;

cout << "\*";

}

}

cout << endl;

return password;

}

Bài 6. Xây dựng hàm nhận vào một chuỗi ký tự có dạng: “ngay-thang-nam” với mỗi giá trị ngày, tháng là 2 ký tự, năm là 4 ký tự (Ví dụ: “06-01-2007”), hàm trả về xem ngày đó là ngày thứ mấy trong năm. Viết chương trình kiểm chứng hàm vừa xây dựng.

#include <iostream>

#include <string>

using namespace std;

int daysInMonth(int month, int year);

int dayOfYear(string date);

#pragma warning( disable : 4996 )

int main() {

string date;

cout << "Nhap vao ngay thang nam: ";

getline(cin, date);

int day = dayOfYear(date);

cout << "The day of year is: " << day << endl;

return 0;

}

int daysInMonth(int month, int year)

{

if (month == 2)

{

if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {

return 29;

}

else

{

return 28;

}

}

else if (month == 4 || month == 6 || month == 9 || month == 11)

{

return 30;

}

else

{

return 31;

}

}

int dayOfYear(string date)

{

int month, day, year;

sscanf(date.c\_str(), "%d-%d-%d", &day, &month, &year);

int totalDays = day;

for (int i = 1; i < month; i++)

{

totalDays += daysInMonth(i, year);

}

return totalDays;

}

Bài 7. Viết chương trình nhập vào chuỗi ngày tháng năm theo định dạng: “thang/ngay/nam”. Sau đó trả về định dạng tương tự nhưng theo qui cách khung giờ quốc tế. Ví dụ: Nhập “07/21/55” hoặc “7/21/1955” Thì kết quả: July 21, 1955

#include <iostream>

#include <string>

using namespace std;

string NgayQT(string s);

#pragma warning( disable : 4996 )

int main()

{

string s;

cout << "Nhap ngay/thang/nam: ";

getline(cin, s);

cout << "Dang ngay quoc te: " << NgayQT(s) << endl;

return 0;

}

string NgayQT(string s)

{

string months[] = { "Jannuary", "February", "March", "April", "May, June", "July", "August", "September", "October", "November", "December" };

int pos = s.find("/");// tim ki tu / dau tien

int thang = atoi(s.substr(0, pos).c\_str());//lay chuoi truoc /, chuyen ve Cstring, chuyen ve dang so

s = s.substr(pos + 1); // bo phan dau va ki tu /

pos = s.find("/");

int ngay = atoi(s.substr(0, pos).c\_str());//lay chuoi truoc /, chuyen ve Cstring, chuyen ve dang so

int nam = atoi(s.substr(pos + 1).c\_str());//lay chuoi sau /, chuyen ve Cstring, chuyen ve dang so

return months[thang - 1] + " " + to\_string(ngay) + ", " + to\_string(nam);

}

Bài 8. Viết hàm nhận vào tham số là ba chuỗi s1, s2, s3. Hàm này thực hiện chức năng tìm kiếm và thay thế tất cả chuỗi con s2 trong s1 bằng chuỗi s3 và trả về chuỗi kết quả.

string Thaythe(string s1, string s2, string s3)

{

int vitri= s1.find(s2,0);

s1.replace(vitri, s2.length(), s3);

return s1;

}

//Bài 9. Xây dựng chương trình nhập vào hai chuỗi s1, s2 và một vị trí x.Chèn chuỗi s2 vào s1 tại vị

//trí x nếu x là vị trí hợp lệ.In chuỗi kết quả ra màn hình.

#include <iostream>

#include <string>

using namespace std;

string Insertx(string s1, string s2, int x);

int main()

{

string s1, s2;

int x;

cout << "Nhap chuoi s1: ";

getline(cin, s1);

cout << "Nhap chuoi s2: ";

getline(cin, s2);

cout << "Nhap vi tri x muon chen: ";

cin >> x;

cout << "Chuoi s1 sau khi them: " << Insertx(s1, s2, x) << endl;

return 0;

}

string Insertx(string s1, string s2, int x)

{

if (x>=0 && x<=s1.length())

{

s1.insert(x, s2);

return s1;

}

}

//Bài 10. Viết hàm mã hóa chuỗi theo nguyên tắc sau :

//-Nếu là chữ cái : thay bằng ký tự đứng sau nó trong bảng mã ascii n ký tự theo hướng xoay

//vòng(với n được nhập từ bàn phím), ví dụ n = 2 : ‘A’ thay bằng ‘C’, ‘B’ thay bằng ‘D’, ‘Y’ thay bằng ‘A’, ‘Z’ thay bằng ‘B’.

//- Nếu là chữ số : thay bằng chữ số cách nó n giá trị theo hướng xoay vòng(với n được nhập từ

// bàn phím), ví dụ n = 2 : ‘0’ thay bằng ‘2’, ‘1’ thay bằng ‘3’, ‘8’ thay bằng ‘0’, ‘9’ thay bằng ‘1’.

// - Viết chương trình kiểm chứng hàm vừa xây dựng, giả sử nhập “Hello128”, n = 2, chuỗi mã hóa là “Jgnnq340”

#include <iostream>

#include <string>

using namespace std;

string Mahoa(string s, int n);

#pragma warning( disable : 4996 )

int main()

{

string s;

int n;

cout << "Nhap chuoi s: ";

getline(cin, s);

cout << "Nhap ma hoa n: ";

cin >> n;

cout << "Chuoi sau khi ma hoa: " << Mahoa(s, n) << endl;

return 0;

}

string Mahoa(string s, int n)

{

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

{

if (isdigit(s[i]))// neu s[i] la so

{

if ((s[i]+n) > '9')

s[i] = s[i] + n - 10;

else

s[i] = s[i] + n;

}

else if (isalpha(s[i])) //neu s[i] la ki tu chu

{

if (islower(s[i]))//la ki tu thuong

{

if ((s[i]+n) > 'z')

s[i] = s[i] + n - 26;

else

s[i] = s[i] + n;

}

else //(s[i])la ki tu hoa

{

if ((s[i]+n) > 'Z')

s[i] = s[i] + n - 26;

else

s[i] = s[i] + n;

}

}

}

return s;

}