

Bài thực hành 10

Bạn không cần xem các bài mẫu nếu đã nắm chắc lý thuyết phần xử lý chuỗi trong C++.

Mục tiêu

- Hiểu và làm việc với 2 kiểu chuỗi ký tự - chuỗi kiểu C là một mảng ký tự và chuỗi kiểu C++ là một biến của lớp string.
- Chuỗi C: các hàm thư viện xử lý chuỗi C (<cstring> và <cctype>)
- Vận dụng chuỗi C thao tác với đối số dòng lệnh
- Chuỗi kiểu C++: các phương thức của lớp string trong thư viện <string>

Bài mẫu

Bài mẫu 1. [noi.cpp] Chương trình ví dụ làm việc với đối số dòng lệnh. In ra màn hình kết quả nối đối số thứ 2 và thứ 3 nếu tổng độ dài không quá 1000.

```
// Neu nguoi dung nhap "noi xau1 xau2"
// thi in ra ket qua noi xau2 vao sau xau1
// Neu nguoi dung nhap it hoac nhieu hon 3 doi so
// thi in ra huong dan su dung
#include <iostream>
#include <cstring>
using namespace std;

const int N_ARGC = 3;
const int MAX_LEN = 1001;
int main(int argc, char* argv[]){
    if(argc != N_ARGC){
        cout << "Cu phap: noi xau1 xau2" << endl
             << "Trong do:" << endl
             << "- noi la ten chuong trinh" << endl
             << "- xau1 la nua dau ket qua" << endl
             << "- xau2 la nua cuoi ket qua" << endl
             << "Luu y: Ket qua noi khong duoc vuot qua "
             << MAX_LEN - 1 << " ki tu." << endl;
        return -1;
    }
    int n = strlen(argv[1]) + strlen(argv[2]) + 1;
    if(n > MAX_LEN){
        cerr << "Ket qua noi vuot qua vuot qua "
             << MAX_LEN - 1 << " ki tu." << endl;
        return -1;
    }

    char ketqua[MAX_LEN] = "";
    strcat(ketqua, argv[1]);
    strcat(ketqua, argv[2]);
    cout << ketqua << endl;

    return 0;
}
```

Kết quả thực thi:

```
>noi ?
Cu phap: noi xau1 xau2
Trong do:
```

```
- noi la ten chuong trinh
- xau1 la nua dau ket qua
- xau2 la nua cuoi ket qua
Luu y: Ket qua noi khong duoc vuot qua 1000 ki tu.
```

```
>noi prog ramming
programming
```

Bài mẫu 2. [cstr.cpp] Chương trình ví dụ làm việc với chuỗi C

```
// Modified from
// http://www.cprogramming.com/tutorial/lesson9.html
#include <iostream> // de su dung cout
#include <cstring> // de su dung cac ham tren xau C

using namespace std;

int main()
{
    char name[50];
    char lastname[50];
    char fullname[100]; // du lon de luu ca name va lastname

    cout << "Moi nhap vao ten ban: ";
    cin.getline(name, 50);
    if(strcmp(name, "Diep") == 0) // so sanh 2 xau C, truong hop bang
        cout << "Ten toi giong ten ban day.\n";
    else // truong hop khac
        cout << "Ten toi khac ten ban.\n";

    // Tinh do dai ten vua nhap
    cout << "Ten ban chua " << strlen(name) << " chu cai.\n";
    cout << "Hay nhap ho cua ban: ";
    cin.getline(lastname, 50);
    fullname[0] = '\0'; // strcat tim den '\0' de noi' vao phia sau
    strcat(fullname, lastname); // sao lastname vao fullname
    strcat(fullname, " "); // them dau cach de tach ho va ten
    strcat(fullname, name); // sao name vao cuoi fullname
    cout << "Ho ten cua ban la: " << fullname << "\n";
    cin.get();
    return 0;
}
```

Kết quả thực thi:

```
Moi nhap vao ten ban: Mai
Ten toi khac ten ban.
Ten ban chua 3 chu cai.
Hay nhap ho cua ban: Tran
Ho ten cua ban la: Tran Mai
```

Bài mẫu 3. [strclass.cpp] Chương trình ví dụ làm việc với các hàm thành viên của lớp string

```
// Note: This is modified from CS Dep C++ manual
#include <iostream>
#include <string>

using namespace std;
```

```

int main()
{
    string string1("cat");
    string string2;
    string string3;

    string2 = string1; // gan string1 vao string2
    string3.assign(string1); // gan string1 vao string3
    cout << "string1: " << string1 << "\nstring2: " << string2
         << "\nstring3: " << string3 << "\n\n";

    // chinh sua string2 va string3
    string2[0] = string3[2] = 'r';
    cout << "Sau khi chinh sua xau thu 2 va xau thu 3:\n"
         << "string1: " << string1
         << "\nstring2: " << string2 << "\nstring3: ";
    // thuc thi ham thanh vien at()
    for(int i = 0; i < string3.length(); i++)
        cout << string3.at(i);

    // khai bao string4
    string string4(string1 + "apult"); // ghep xau

    // phep += nap chong
    string3 += "pet"; // tao ra xau "carpet"
    string1.append("acomb"); // tao ra xau "catacomb"

    string string5;
    // chen string1 vao string5 sau khi loai bo 4 ky tu dau cua string1
    // de tao ra xau "comb" (string5 da duoc khoi tao rong luc ban dau)
    string5.append(string1, 4, string1.length() - 4);

    cout << "\n\nSau khi ghep:\nstring1: " << string1
         << "\nstring2: " << string2 << "\nstring3: " << string3
         << "\nstring4: " << string4 << "\nstring5: " << string5 << endl;

    cin.get();
    return 0;
} // ket thuc ham main

```

Kết quả thực thi:

```

string1: cat
string2: cat
string3: cat

```

Sau khi chinh sua xau thu 2 va xau thu 3:

```

string1: cat
string2: rat
string3: car

```

Sau khi ghep:

```

string1: catacomb
string2: rat
string3: carpet
string4: catapult
string5: comb

```

Bài tập

Lưu ý: Trong lời giải của câu 1-4, bạn chỉ được dùng xâu C. Câu 5 phải dùng lớp string.

Câu 1. [reverse.cpp]

Hãy viết hàm đảo ngược nội dung của 1 xâu. Đọc từ bàn phím không quá 50 xâu (xâu có thể chứa dấu trắng ở giữa), đảo ngược nội dung của từng xâu rồi ghi ra màn hình.

Ví dụ:	Input:	Output:
	<code>nhap xau: ik auig iht</code>	<code>thi giua ki</code>
	<code>nhap tien (c/k)?c</code>	<code>lop k54cd</code>
	<code>nhap xau: dc45k pol</code>	
	<code>nhap tien (c/k)?k</code>	

Câu 2. [stdname.cpp]

Nhập vào từ bàn phím họ tên một người. Chuẩn hóa xâu này với ba bước sau:

- Loại bỏ dấu cách ở hai đầu xâu.
- Loại bỏ dấu cách thừa ở giữa các từ.
- Viết hoa các chữ cái đầu từ.

Ví dụ:	Input:
	<code>Nhap mot xau: nguyen Viet anh</code>
	Output:
	<code>Sau khi bo dau cach o hai dau:</code>
	<code>nguyen Viet anh</code>
	<code>Sau khi loai bo dau cach giua cac tu:</code>
	<code>nguyen Viet anh</code>
	<code>Sau khi viet hoa cac chu cai dau cac tu:</code>
	<code>Nguyen Viet Anh</code>

Câu 3. [count.cpp]

Nhập vào một xâu kí tự chỉ gồm các chữ cái và dấu cách. Thống kê số lần xuất hiện của từng chữ cái (không phân biệt viết hoa viết thường) trong xâu, sắp xếp chúng theo thứ tự giảm dần và in kết quả.

Ví dụ: Chương trình hoạt động như sau:

Moi ban nhap mot xau:	Tan suat cua cac ky tu:
<code>We are the world</code>	<code>e:3</code>
	<code>w:2</code>
	<code>r:2</code>
	<code>a:1</code>
	<code>t:1</code>
	<code>h:1</code>
	<code>o:1</code>
	<code>l:1</code>
	<code>d:1</code>

Câu 4. [caesar.cpp]

Mã hóa Caesar được xây dựng như sau:

Với xâu kí tự s và một số tự nhiên k , mỗi kí tự bị mã hóa bằng cách dịch về sau k vị trí trong bảng chữ cái (nếu hết bảng chữ cái thì quay ngược trở lại). Việc giải mã được thực hiện ngược lại với việc mã hóa.

Hãy viết chương trình nhận 4 đối số dòng lệnh

1. Tên chương trình: caesar
2. Bằng “encoding” nếu muốn mã hóa, “decoding” nếu muốn giải mã
3. k
4. Xâu s (đặt xâu này trong cặp ngoặc kép nếu chứa dấu cách)

Sau đó in ra xâu mã hóa hoặc giải mã Caesar của s tùy theo đối số thứ 2. Ở đây ta sử dụng bảng chữ cái tiếng Anh.

Ví dụ:

```
>caesar encoding 3 "Nguyen Viet Anh"
Qjxbhq Ylhw Dqk

>caesar decoding 3 "Qjxbhq Ylhw Dqk"
Nguyen Viet Anh
```

Câu 5. [student.cpp]

Cho cấu trúc Date biểu diễn ngày tháng và cấu trúc Student biểu diễn thông tin sinh viên như sau:

```
struct Date{
    int day;           // ngay
    int month;        // thang
    int year;         // nam
};
struct Student{
    int id;           // ma sinh vien
    string fullname; // ho ten
    Date birthday;   // ngay sinh
    string yclass;   // lop khoa hoc
};
```

Viết chương trình cho người dùng chọn 1 trong 4 tính năng từ menu:

1. Nhập sinh viên: *nhập thêm vào danh sách sinh viên đến khi người dùng không muốn nhập tiếp*
2. Sắp xếp theo họ tên: *sắp xếp tăng dần danh sách sinh viên theo tên, nếu tên trùng nhau thì sắp xếp theo họ và đệm). Sau đó in ra danh sách đã sắp.*
3. Sắp xếp theo lớp khóa học: *sắp xếp tăng dần danh sách sinh viên theo lớp khóa học. Sau đó in ra danh sách đã sắp.*
4. Thoát

Sau khi thực hiện xong 1 tính năng (1-3) thì in lại menu cho phép tiếp tục chọn.

Yêu cầu cài đặt: Hãy sử dụng `vector<Student>` để lưu danh sách. Trong câu này bạn không được sử dụng hàm sắp xếp có sẵn trong các thư viện của C++.