



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN  
HỆ CHÍNH QUY  
MÔN: **NHẬP MÔN LẬP TRÌNH**

## **HƯỚNG DẪN THỰC HÀNH TUẦN 09**

### **KIỂU KÝ TỰ VÀ KIỂU CHUỖI**

## MỤC LỤC

1	Các ví dụ.....	3
1.1	Ví dụ 1: .....	3
1.2	Ví dụ 2 .....	4
1.3	Ví dụ 3 .....	5
2	Bài tập .....	7

# 1 Các ví dụ

## 1.1 Ví dụ 1:

Yêu cầu: Nhập một chuỗi S từ bàn phím. Kiểm tra xem chuỗi có phải là chuỗi đối xứng. Ví dụ:

Nhập: S = "aBCdCBa"

Xuất: Đối xứng

Nhập: S = "aBCdBCa"

Xuất: Không đối xứng

Ý tưởng: Giả sử chuỗi là chuỗi đối xứng. Nếu phát hiện 1 vị trí mà không đối xứng với nó (cùng vị trí nhưng tính từ cuối chuỗi) thì chuỗi không còn đối xứng nữa.

Chương trình:

```
#include <string.h>
#include "stdio.h"
#include "conio.h"

// Nền định nghĩa MAX để có thể thay đổi kích thước mảng nhanh chóng
#define MAX 100
// Nền đặt prototype ở đây để biết hàm nhận vào cái gì và trả về cái gì
// Đầu vào: Chuỗi cần kiểm tra
// Đầu ra: 1 nếu là chuỗi đối xứng, 0 nếu là chuỗi không đối xứng
int LaChuoiDoiXung(char []);

void main()
{
    char str[MAX];
    printf("Nhap 1 chuoi: ");
    // Nền sử dụng gets thay vì scanf để có thể nhập chuỗi có khoảng trắng
    gets(str);
    int kq = LaChuoiDoiXung(str);
    if (kq==1)
        printf("Chuoi \"%s\" la chuoi doi xung.", str); // Sử dụng \" để xuất "
    else
        printf("Chuoi \"%s\" khong phai la chuoi doi xung.", str);
    getch();
}

int LaChuoiDoiXung(char str[MAX])
{
    int Flag = 1;    // cứ giả sử đây là chuỗi đối xứng
    int l = strlen(str); // strlen để lấy độ dài chuỗi. Nền tính 1 lần để sử dụng lại
    for (int i=0; i<l/2; i++)
        if (str[i]!=str[l-i-1])
            Flag = 0; // phát hiện không đối xứng thì cập nhật cờ
    return Flag;
}
```

```
}
```

Ghi chú: có thể không cần sử dụng biến Flag để kiểm tra mà có thể return 0; ngay khi kiểm tra thấy có vị trí không còn đối xứng. Cụ thể như sau:

```
int LaChuoIdoiXung(char str[MAX])
{
    int l = strlen(str); // strlen để lấy độ dài chuỗi. Nên tính 1 lần để sử dụng lại
    for (int i=0; i<l/2; i++)
        if (str[i]!=str[l-i-1])
            return 0; // phát hiện không đối xứng thì trả về 0 ngay!
    return 1; // chưa lần nào phát hiện vị trí không đối xứng trong chuỗi đối xứng
}
```

## 1.2 Ví dụ 2

Yêu cầu: Nhập một chuỗi S từ bàn phím. Tìm ký tự xuất hiện nhiều nhất trong chuỗi đó và số lần xuất hiện.

Ý tưởng:

- Dùng 2 biến lưu ký tự xuất hiện nhiều nhất và số lần xuất hiện nhiều nhất đó.
- Ban đầu ký tự xuất hiện nhiều nhất chưa có nên số lần xuất hiện nhiều nhất là 0.
- Xét 1 ký tự tại vị trí i. Đếm xem từ vị trí i về sau ký tự này xuất hiện bao nhiêu lần. Nếu số lần xuất hiện này nhiều hơn số lần hiện tại thì ta cập nhật ký tự xuất hiện nhiều nhất và số lần xuất hiện nhiều nhất trùng với ký tự vừa xét.

Chương trình:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
// Nên định nghĩa MAX để có thể thay đổi kích thước mảng nhanh chóng
#define MAX 100
// Nên đặt prototype ở đây để biết hàm nhận vào cái gì và trả về cái gì
// Đầu vào: Chuỗi cần kiểm tra và biến chứa ký tự xuất hiện nhiều nhất tìm được
// Đầu ra: Số lần xuất hiện nhiều nhất của ký tự. Bằng 0 nếu không có ký tự nào
int KyTuXuatHienNhiềuNhat(char [], char &);
void main()
{
    char str[MAX];
    printf("Nhập 1 chuỗi: ");
    // Nên sử dụng gets thay vì scanf để có thể nhập chuỗi có khoảng trắng
    gets(str);
    char chr;
    int max = KyTuXuatHienNhiềuNhat(str, chr);
    if (max!=0) // Trường hợp max khác 0 ⇨ Chuỗi không rỗng
        printf("Ky tu %c xuất hiện nhiều nhất là %d lần.", chr, max);
}
```

```

    else
        printf("Chuoi rong! Khong co ky tu xuat hien nhieu nhat.");
        getch();
}

int KyTuXuatHienNhiềuNhat(char str[], char &chr)
{
    int i, j, length = strlen(str);
    char curchr; // Ký tự đang xét
    int curcount; // Số lần xuất hiện của ký tự đang xét
    int max = 0; // Số lần xuất hiện nhiều nhất ban đầu là 0 (chưa có)
    for (i=0; i<length; i++)
    {
        curchr = str[i];
        curcount = 1;
        // Lấy ký tự thứ i ra kiểm tra với các ký tự sau i
        for (j=i+1; j<length; j++)
            if (str[j] == str[i])
                curcount++;
        // Tìm được số lần xuất hiện nhiều hơn max thì cập nhật lại số lần và ký tự
        if (max < curcount)
        {
            max = curcount;
            chr = curchr;
        }
    }
    return max;
}

```

### 1.3 Ví dụ 3

Yêu cầu: Nhập họ tên của một người từ bàn phím. Hãy chuẩn hóa chuỗi họ tên này. (Xóa các khoảng trắng thừa và ký tự đầu tiên của họ, chữ lót và tên phải viết hoa, các ký tự còn lại viết thường). Ví dụ:

Nhập: " NgUyen VaN A "

Xuất: "Nguyen Van A"

Ý tưởng:

- Xóa các khoảng trắng dư ở bên trái chuỗi.
- Xóa các khoảng trắng dư ở bên phải chuỗi.
- Nếu tìm được 2 ký tự khoảng trắng dính nhau trong chuỗi thì xóa bớt 1 khoảng trắng.
- Cho tất cả thành chữ thường.
- Cập nhật ký tự đầu tiên và các ký tự mà trước nó là khoảng trắng thành chữ hoa.

## Chương trình:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
// Nên định nghĩa MAX để có thể thay đổi kích thước mảng nhanh chóng
#define MAX 200
// Nên đặt prototype ở đây để biết hàm nhận vào cái gì và trả về cái gì
// Đầu vào: Chuỗi cần chuẩn hóa
// Đầu ra: không có (chuỗi đưa vào được cập nhật sau khi chuẩn hóa)
void ChuanHoaChuoi(char []);
void main()
{
    char str[MAX];
    printf("Nhap ho ten: ");
    // Nên sử dụng gets thay vì scanf để có thể nhập chuỗi có khoảng trắng
    gets(str);
    ChuanHoaChuoi(str);
    printf("Chuoi sau khi chuan hoa: %s", str);
    getch();
}
void ChuanHoaChuoi(char str[])
{
    int i;
    // Xóa các khoảng trắng dư bên trái
    // Nếu ký tự đầu tiên vẫn là khoảng trắng thì dịch chuyển chuỗi qua trái 1 ký tự
    while (str[0]==' ')
    {
        for (i=0; i<strlen(str)-1; i++)
            str[i] = str[i+1];
        str[strlen(str)-1] = '\0'; // Cập nhật lại ký tự kết thúc chuỗi lùi lại
    }
    // Xóa các khoảng trắng dư bên phải
    // Nếu ký tự cuối cùng vẫn là khoảng trắng thì dời lùi ký tự kết thúc chuỗi 1 ký tự
    while (str[strlen(str)-1]==' ')
    {
        str[strlen(str)-1] = '\0';
    }
    // Xóa các khoảng trắng dư ở giữa
    // Nếu tìm thấy khoảng trắng và ký tự kế tiếp cũng là khoảng trắng
    // Thì xóa khoảng trắng thừa đó (dời chuỗi qua trái 1 ký tự tại khoảng trắng đó)
    i = 0;
    while (i<strlen(str)-1)
    {
        if (str[i]==' ')
            if (str[i+1]==' ')
            {
                for (int j=i+1; j<strlen(str)-1; j++)
                    str[j] = str[j+1];
                str[strlen(str)-1] = '\0';
            }
            else
                i++;
        else
            i++;
    }
    // Biến tất cả ký tự của chuỗi thành chữ thường
```

```

strlwr(str);
// Chữ đầu tiên là chữ hoa
str[0] = str[0]-32;
// Chữ sau khoảng trắng sẽ được đổi thành chữ hoa
for (i=0; i<strlen(str)-1; i++)
    if (str[i]==' ')
        str[i+1] = str[i+1]-32;
}

```

### Chú ý:

- Có thể viết 1 hàm xóa 1 ký tự tại vị trí x trong chuỗi để sử dụng chung (xem như Bài tập)
- Ký tự kết thúc chuỗi là ký tự “\0”
- Do ký tự thường đứng sau ký tự hoa 32 ký tự trong bảng mã ASCII nên muốn đổi ký tự thường sang hoa thì ta trừ giá trị của ký tự thường với 32. Đơn giản hơn là sử dụng hàm toupper trong thư viện ctype.h.

## 2 Bài tập

**Bài 1:** Viết chương trình nhập chuỗi ký S và thực hiện các thao tác sau:

- a. Đếm và cho biết số lượng khoảng trắng, số lượng ký số, số lượng chữ cái latin, số lượng các ký tự khác.
- b. Đếm và cho biết số lượng từ của chuỗi – các từ cách nhau bởi khoảng trắng.
- c. Tìm từ ngắn nhất và dài nhất.
- d. Biến đổi chuỗi sao cho các ký tự đầu mỗi từ là ký tự in hoa, các ký tự khác in thường.

**Bài 2:** Viết chương trình nhập một số nguyên, xuất lại số đó ở dạng chuỗi nhưng có dấu “,” ngăn cách hàng triệu, ngàn... Ví dụ:

Nhập: N = 123456789

Xuất: S = “123,456,789”

**Bài 3:** Không sử dụng các hàm có sẵn. Viết chương trình xóa n ký tự tại vị trí i trong chuỗi S. Ví dụ:

Nhập: S = “Nguyen Van A” i = 2 n = 3 (Xóa 3 ký tự từ ký tự 2 trong chuỗi S)

Xuất: S = “Nen Van A”

**Bài 4:** Nhập một chuỗi S từ bàn phím và một ký tự C. Đếm xem ký tự C xuất hiện bao nhiêu lần trong chuỗi S đó. Ví dụ:

Nhập: "Nguyen Van A" C = "u"

Xuất: 1 lần

**Bài 5:** Viết các hàm nhập vào từ bàn phím danh sách học sinh một lớp, sắp xếp lại danh sách theo thứ tự abc của Tên, nếu trùng Tên thì sắp xếp theo thứ tự abc của Họ.

**Bài 6:** Viết chương trình nhập từ bàn phím 2 xâu ký tự S1 và S2. Hãy xét xem S1 có xuất hiện bao nhiêu lần trong S2 (hoặc ngược lại S2 xuất hiện bao nhiêu lần trong S1) và tại những vị trí nào?

**Bài 7:** Không sử dụng các hàm có sẵn. Viết chương trình chèn chuỗi S2 vào chuỗi S1 tại vị trí i trong chuỗi S1. Ví dụ:

Nhập: S1 = "Nguyen Van A"

S2 = "Le "

i = 8 (Chèn chuỗi S2 vào chuỗi S1 tại vị trí 8)

Xuất: S1 = "Nguyen LeVan A"

**Bài 8:** Nhập một chuỗi S từ bàn phím. Đếm xem có bao nhiêu từ có nhiều hơn n ký tự có trong chuỗi S.

Nhập: "Nguyen Van A" n = 2

Xuất: 2 từ