



IT001 - NHẬP MÔN LẬP TRÌNH

CHƯƠNG 7.3: CHUỖI

Trong ngôn ngữ lập trình C++, chuỗi ký tự (strings) đóng vai trò thiết yếu trong việc xử lý và thao tác dữ liệu văn bản, cung cấp một cách tiện lợi và hiệu quả để làm việc với các chuỗi văn bản từ đơn giản đến phức tạp. Chương này chúng ta sẽ tìm hiểu về chuỗi C-string và các thao tác trên chuỗi.

Khoa Khoa học Máy tính



NỘI DUNG

7.1 Giới thiệu mảng

7.2 Khái niệm mảng

7.3 Các yếu tố xác định mảng

7.4 Mảng một chiều

7.5. Mảng hai chiều

7.6 Mảng nhiều chiều

7.7 Chuỗi C-string



7.7 Chuỗi C-string



7.7 Chuỗi C-string

7.7.1 Khái niệm chuỗi

7.7.2 Khai báo, khởi tạo chuỗi

7.7.3 Nhập xuất chuỗi

7.7.4 Phép gán trên chuỗi

7.7.5 Truyền chuỗi cho hàm và lời gọi hàm

7.7.6 Một số thao tác trên chuỗi

7.7.7 Một số hàm thông dụng trong thư viện chuỗi



7.7 Chuỗi C-string

7.7.1 Khái niệm chuỗi



7.7.1 Khái niệm chuỗi

- Kiểu **char** chỉ chứa được một ký tự. Để lưu trữ một chuỗi (nhiều ký tự) ta sử dụng mảng (một chiều) các ký tự.
- Chuỗi ký tự kết thúc bằng ký tự '\0' (null)
- Độ dài chuỗi = kích thước mảng - 1 (trừ đi ký tự kết thúc chuỗi)
- Ví dụ:

char Hoten[30]; // Độ dài 29 ký tự + 1 ký tự kết thúc chuỗi

char NgaySinh[9]; // Độ dài 8 ký tự + 1 ký tự kết thúc chuỗi

char *sName; // khai báo chuỗi dùng con trỏ (sẽ học ở



7.7 Chuỗi C-string

7.7.2 Khai báo, khởi tạo chuỗi

7.7.2 Khởi tạo chuỗi ký tự

- Khởi tạo chuỗi như khởi tạo mảng một chiều các ký tự.
- Độ dài chuỗi:

```
char s[10] = {'N', 'M', 'L', 'T', ' ', '1', '\\0'};
```

```
char s[10] = "NMLT 1"; // Tự động thêm '\\0'
```

0	1	2	3	4	5	6	7	8	9
N	M	L	T		1	\\0			

- Tự xác định độ dài:

```
char s[] = {'N', 'M', 'L', 'T', ' ', '1', '\\0'};
```

```
char s[] = "NMLT 1"; // Tự động thêm '\\0'
```

0	1	2	3	4	5	6
N	M	L	T		1	\\0



7.7 Chuỗi C-string

7.7.3 Nhập xuất chuỗi



7.7.3 Nhập xuất chuỗi

- Hàm nhập chuỗi: (Xem thêm trong Chương 3 phần Nhập xuất)
 - `cin >>`, `cin.get`, **`cin.getline`**
 - `fgets`, `scanf`
- Hàm xuất chuỗi: **`cout <<`**, `printf`, `puts`
- Hàm tự động thêm ký tự NULL (**`'\0'`**) vào cuối biến chuỗi.
- Ví dụ:

```
char str[100];
```

```
cin >> str; // Không nhận khoảng trắng
```

```
fgets(str, 100, stdin); // Ký tự \n sẽ được đưa vào cuối chuỗi
```

```
cin.get(str, 100); // Ký tự \n sẽ không mất đi trên Steam
```

```
cin.getline(str, 100);
```



Ví dụ: Nhập xuất chuỗi

```
#include <stdio.h>
#include <iostream>
#include <limits>
using namespace std;
int main () {
    char str1[100], str2[100], str3[100], str4[100];
    cout << "Nhập str1: "; cin >> str1; // Không nhận khoảng trắng
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    cout << "Nhập str2: "; fgets(str2, 100, stdin); // Ký tự \n sẽ được đưa vào cuối chuỗi

    cout << "Nhập str3: "; cin.get(str3, 100); // Ký tự \n sẽ không mất đi trên Steam
    cin.ignore();

    cout << "Nhập str4: "; cin.getline(str4, 100);

    cout << "\nXuất str1: " << str1 << endl;
    puts(str2);
    puts(str3);
    cout << "Xuất str4: " << str4 << endl;
    return 0;
}
```

Kết quả thực thi:

Nhap str1: chuoi thu nhat
Nhap str2: chuoi thu hai
Nhap str3: chuoi thu ba
Nhap str4: chuoi thu tu

Xuat str1: chuoi
Xuat str2: chuoi thu hai
Xuat str3: chuoi thu ba
Xuat str4: chuoi thu tu



7.7 Chuỗi C-string

7.7.4 Phép gán trên chuỗi



7.7.4 Phép gán trên chuỗi

- Các phép gán sai:

```
char s[10];  
s = "hello";
```

```
char myword[10];  
myword[] = "hello";
```

```
char s[10];  
s={'h', 'e', 'l', 'l', 'o',  
'\0'};
```

- Các phép gán và khởi tạo đúng:

```
char s[10];  
s[0] = 'h';  
s[1] = 'e';  
s[2] = 'l';  
s[3] = 'l';  
s[4] = 'o';  
s[5] = '\0';
```

```
char *pTen;  
pTen = "hello";
```

```
char *pTen = "hello";
```

```
char pTen[] = "hello";
```



Ví dụ: Cho biết kết quả đoạn code sau và so sánh

```
#include <iostream>
using namespace std;
#define MAX 300
char *getValue(){
    char *temp="NMLT";
    return temp;
}
char *getValue1(){
    char temp[]="CTDL&GT";
    return temp;
}
```

```
int main() {
    char *d;
    d=getValue();
    char *e;
    e = getValue1();
    cout << "d: " << d <<
endl;
    cout << "e: " << e <<
endl;
    return 0;
}
```



7.7 Chuỗi C-string

7.7.5 Truyền chuỗi cho hàm và lời gọi hàm



7.7.5 Truyền chuỗi cho hàm và lời gọi hàm

- Truyền tham số chuỗi cho hàm cũng giống như truyền tham số cho biến mảng 1 chiều.
- Ví dụ:

```
void NhapChuoi(char s[100]);  
void NhapChuoi(char s[]);  
void NhapChuoi(char *s);
```




7.7.5 Truyền chuỗi cho hàm và lời gọi hàm

- Ví dụ:

```
#include <iostream>
using namespace std;
void NhapChuoi(char *s){
    cout << "Nhap str: "; cin.getline(s, 100);
}
void XuatChuoi(char s[]){
    cout << "Xuat str: " << s << endl;
}
int main () {
    char str[100];
    NhapChuoi(str);
    XuatChuoi(str);
    return 0;
}
```



7.7 Chuỗi C-string

7.7.6 Các thao tác trên chuỗi



7.7.6 Các thao tác trên chuỗi

1. Đếm các ký tự khoảng trắng trong chuỗi ký tự
2. Đếm các ký tự hoa / thường trong chuỗi ký tự
3. Đổi các từ ở đầu câu sang chữ hoa và những từ không phải đầu câu sang chữ thường.
4. Chuyển các ký tự viết hoa thành viết thường
5. Chuyển các ký tự viết thường thành viết hoa
6. Liệt kê các từ trong chuỗi
7. Xóa các khoảng trắng đầu chuỗi / cuối chuỗi
8. Kiểm tra chuỗi đối xứng



1. Đếm các ký tự khoảng trắng

```
int DemKhoangTrang(char chuo[i][100]) {  
    int dem = 0;  
  
    for (int i = 0; i < strlen(chuo[i]); i++)  
        if (chuo[i][i] == ' ') dem++;  
  
    return dem;  
}
```




2. Đếm các ký tự hoa / thường

```
void DemKTThuong(char chuoai[]) {  
    int i, dt = 0, dh = 0;  
  
    for (i = 0; i < strlen(chuoai); i++)  
        if ((chuoai[i] >= 'a') && (chuoai[i] <= 'z'))  
            dt++;  
        else if ((chuoai[i] >= 'A') && (chuoai[i] <= 'Z'))  
            dh++;  
  
    printf("So ky tu thuong : % d", dt);  
    printf("So ky tu hoa : % d", dh);  
}
```



3. Đổi hoa - thường

```
void DoiHoaThuong(char chuo[i][100]) {  
    chuo[i][0] = toupper(chuo[i][0]);  
  
    for (int i = 1; i < strlen(chuo[i]); i++)  
        chuo[i][i] = tolower(chuo[i][i]);  
  
    printf("Xuat chuo[i]: %s", chuo[i]);  
}
```



4. Chuyển các ký tự viết hoa thành viết thường

```
void ChuyenHoaSangThuong(char chuo[i][100]) {  
    char kq[100];  
    strcpy(kq, chuo[i]);  
    for (int i = 0; kq[i] != '\0'; i++)  
        if ((kq[i] >= 'A') && (kq[i] <= 'Z'))  
            kq[i] = tolower(kq[i]);  
    printf("Xuat chuo[i]: ");  
    puts(kq);  
}
```



5. Chuyển các ký tự viết thg thành viết hoa

```
void ChuyenThuongSangHoa(char chuoì[100]) {  
    char kq[100];  
    strcpy(kq, chuoì);  
  
    for (int i = 0; kq[i] != '\\0'; i++)  
        if ((kq[i] >= 'a') && (kq[i] <= 'z'))  
            kq[i] = toupper(kq[i]);  
  
    printf("Xuat chuoì: ");  
    puts(kq);  
}
```




7. Liệt kê các từ trong chuỗi

```
void LietKeTu(char chuoi[100]) {  
    int d = 0;  
    for (int i = 0; i < strlen(chuoi); i++)  
        if (chuoi[i] == ' ') {  
            for (int j = d; j < i; j++)  
                d = i + 1;  
            printf("%c", chuoi[j]);  
            printf("\n");  
        }  
}
```



7. Xóa các khoảng trắng

```
void xoadau(char str[100]) {  
    while (str[0] == ' ')  
        for (int i = 0; i < strlen(str); ++i)  
            str[i] = str[i + 1];  
}
```

```
void xoacuo(char chuo[i][100]) {  
    while (chuo[i][strlen(chuo[i])] == ' ')  
        chuo[i][strlen(chuo[i])] = '\\0';  
}
```



8. Kiểm tra chuỗi đối xứng

```
int DoiXung(char s[]) {  
    int n = strlen(s);  
    for (int i = 0; i <= (n-1)/2; i++)  
        if (s[i] != s[n - i - 1])  
            return false;  
    return true;  
}
```



7.7 Chuỗi C-string

7.7.7 Một số hàm thông dụng trong thư viện chuỗi



7.7.8 Một số hàm thông dụng trong thư viện

Một số hàm thuộc thư viện `<string.h>`

- `strlen`: hàm tính độ dài chuỗi ký tự
- `strcpy`: hàm sao chép chuỗi ký tự
- `strdup`: hàm tạo bản sao
- `strlwr`: hàm chuyển chuỗi thành chuỗi viết thường
- `strupr`: hàm chuyển chuỗi thành chuỗi viết hoa
- `strrev`: hàm đảo ngược
- `stricmp`: hàm so sánh 2 chuỗi có phân biệt hoa thường
- `stricmp`: hàm so sánh 2 chuỗi không phân biệt hoa thường
- `strcat`: hàm nối 2 chuỗi
- `strstr`: hàm tìm chuỗi trong chuỗi

Hàm tính độ dài chuỗi ký tự: strlen

```
size_t strlen( const char * str );
```



Trả về



- Tính độ dài chuỗi `str`
- `size_t` thay cho `unsigned` (trong `<stddef.h>`) dùng để đo các đại lượng không dấu.

- Độ dài chuỗi `str`

```
char s[] = "Nhap mon lap trinh";  
int len = strlen(s);    // => 18
```


Hàm sao chép chuỗi ký tự: strcpy

```
char *strcpy(char dest[], const char src[])
```



- Sao chép chuỗi src sang chuỗi dest, dừng khi ký tự kết thúc chuỗi '\0' vừa được chép.
- dest phải đủ lớn để chứa src.



- Địa chỉ chuỗi dest



```
char s[100];  
s = "Nhap mon lap trinh"; // sai  
strcpy(s, "Nhap mon lap trinh"); //  
đúng
```

Hàm tạo bản sao: strdup

```
char *strdup(const char str[])
```



- Tạo bản sao của một chuỗi `str` cho trước.
- Hàm sẽ tự tạo vùng nhớ đủ chứa chuỗi `str`.

Trả về



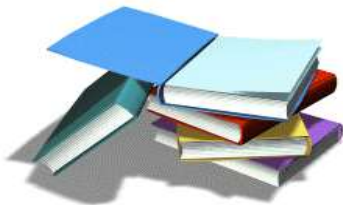
- Thành công: Địa chỉ chuỗi kết quả
- Thất bại: `NULL`

```
char *s;  
s = strdup("Nhap mon lap trinh");
```

hàm chuyển chuỗi thành chuỗi viết thường: `strlwr`



```
char *strlwr( char *str );
```



Chuyển chuỗi `str` thành chuỗi thường ('A' thành 'a', 'B' thành 'b', ..., 'Z' thành 'z')

Trả về

➤ Địa chỉ chuỗi `str`



```
char s[] = "NHAP MON LAP TRINH!!!";  
strlwr(s);  
puts(s);           // nhap mon lap  
trinh!!!
```



Hàm chuyển chuỗi thành chuỗi viết hoa:strupr

`char *strupr(char *str)`



Chuyển chuỗi `str` thành chuỗi viết in ('a' thành 'A', 'b' thành 'B', ..., 'z' thành 'Z')

Trả về

➤ Địa chỉ chuỗi `str`



```
char s[] = "Nhap mon lap trinh!!!";  
strupr(s);  
puts(s);  
TRINH!!!
```

// NHAP MON LAP

Hàm đảo ngược: `strrev`

`char *strrev(char *s)`



Trả về



- Đảo ngược thứ tự các ký tự trong chuỗi (trừ ký tự kết thúc chuỗi)

- Địa chỉ chuỗi kết quả

```
char s[] = "Nhap mon lap trinh!!!";  
strrev(s);  
puts(s);  
pahn
```

// !!!hnirt pa1 nom

Hàm so sánh 2 chuỗi có phân biệt hoa thường: strcmp

```
int strcmp(const char *s1, const char *s2)
```



Trả về



- So sánh hai chuỗi s1 và s2 (phân biệt hoa thường)

- < 0 nếu $s1 < s2$
- $== 0$ nếu $s1 == s2$
- > 0 nếu $s1 > s2$

```
char s1[] = "Nhap mon lap trinh!!!";  
char s2[] = "nhap mon lap trinh!!!";  
int kq = strcmp(s1, s2);  
// => kq < 0
```



Hàm so sánh 2 chuỗi không phân biệt hoa thường: stricmp

```
int stricmp(const char *s1, const char *s2)
```



- So sánh hai chuỗi s1 và s2 (không phân biệt hoa thường)

Trả về



- < 0 nếu $s1 < s2$
- $== 0$ nếu $s1 == s2$
- > 0 nếu $s1 > s2$

```
char s1[] = "Nhap mon lap trinh!!!";  
char s2[] = "NHAP MON LAP TRINH!!!";  
int kq = stricmp(s1, s2);  
// => kq == 0
```

Hàm nối 2 chuỗi: strcat

```
char* strcat(char *dest, const char *src)
```



- Nối chuỗi `src` vào sau chuỗi `dest`.
- Chuỗi `dest` phải đủ chứa kết quả

Trả về

- Địa chỉ của chuỗi được nối



```
char s1[100] = "Nhap mon";  
char s2[] = "lap trinh";  
strcat(s1, " "); // => "Nhap mon "  
strcat(s1, s2); // => "Nhap mon lap trinh"
```

Hàm tìm chuỗi trong chuỗi: strstr

```
char* strstr(const char *s1, const char *s2)
```



Trả về



- Tìm vị trí xuất hiện đầu tiên của s2 trong s1

- Thành công: trả về con trỏ đến vị trí xuất hiện đầu tiên của s2 trong s1.
- Thất bại: trả về NULL

```
char s1[] = "Nhap mon lap trinh";  
char s2[] = "mon";  
if (strstr(s1, s2) != NULL) cout << "Tim  
thay!";
```



Ví dụ: Hàm strcpy

```
#include<iostream>
#include<string.h>
using namespace std;
int main() {
    char s1[] = "Nhap mon lap trinh";
    char s2[100];
    // s2=s1; ==> SAI
    strcpy(s2, s1);
    cout << s1 << endl;
    cout << s2;
    return 0;
}
```

Kết quả thực thi:

Nhap mon lap trinh
Nhap mon lap trinh



Ví dụ: Hàm strstr

```
#include<iostream>
#include<string.h>
using namespace std;

int main() {
    char s1[] = "Nhap mon lap trinh";
    char s2[] = "mon";
    char* s=strstr(s1, s2);
    cout << s;
    return 0;
}
```

Kết quả thực thi:

mon lap trinh



Ví dụ:

```
#include<iostream>
#include<string.h>
using namespace std;
int main() {
    char str1[] = "Nhap mon lap trinh";
    char str2[40], str3[40];

    /* Sao chép chuỗi str1 cho str2*/
    strncpy(str2, str1, sizeof(str2));

    /* Sao chép 5 ký tự chuỗi str2 cho str3 */
    strncpy(str3, str2, 5);
    str3[5] = '\0';    /* null character manually added */

    cout << str1 << endl;
    cout << str2 << endl ;
    cout << str3;

    return 0;
}
```

Kết quả thực thi:

Nhap mon lap trinh
Nhap mon lap trinh
Nhap



Ví dụ:

```
#include<iostream>
#include<string.h>
using namespace std;
int countCharInString(char c, char s[]) {
    int count=0;
    char *pch = strchr(s, c);
    while(pch) {
        count++;
        cout << "found at " << pch - s + 1 << endl;
        pch = strchr(pch + 1, c);
    }
    return count;
}
int main(){
    char s[100]="Nhap mon lap trinh";
    int count = countCharInString('a', s);
    cout << "Ky tu \'a\' xuất hiện " << count << " lần.";
    return 0;
}
```

Kết quả thực thi:

found at 3
found at 11
Ky tu 'a' xuất hiện 2 lần.



Bài tập

Bài 1: Xem thêm một số hàm khác như

- `atoi`, `atol`, `atof` : đổi chuỗi thành số
- `itoa`, `ltoa`, `ultoa`: đổi số thành chuỗi
- `strtok`

a. Nhập vào một số nguyên `i`, biến đổi số này thành chuỗi `s` và ngược lại (dùng hàm `itoa` và `atoi`).

b. Dựa vào khoảng trắng, tách một chuỗi thành các từ sử dụng hàm `strtok`

Bài 2: Viết hàm `upper(char s[])` đổi toàn bộ các ký tự sang ký tự hoa (giống hàm `strupr`)

Bài 3: Viết hàm `lower(char s[])` đổi toàn bộ các ký tự sang ký tự thường (giống hàm `strlwr`)



Bài tập

Bài 4: Viết hàm `proper(char s[])` đổi các ký tự đầu tiên của mỗi từ sang ký tự hoa.

Bài 5: Viết chương trình nhập vào một chuỗi `s` và một ký tự `c`, cho biết số lần `c` xuất hiện trong `s` (dùng hàm `strchr`).

Bài 6: Viết chương trình nhập vào một chuỗi, cho biết:

- Độ dài của chuỗi.
- Trong chuỗi có bao nhiêu ký tự 'a'.
- Cho biết trong chuỗi có bao nhiêu khoảng trắng (dùng hàm `isspace`), ở những vị trí nào.



Bài tập

Bài 7: Nhập vào chuỗi S1 và S2, in ra màn hình:

- Hai chuỗi này có bằng nhau không?
- Hai chuỗi này có bằng nhau không (nếu không phân biệt ký tự hoa và thường)?
- Hai chuỗi này có bằng nhau không nếu chỉ xét 3 ký tự đầu (dùng hàm `strncmp`)?
- Giả sử S1 và S2 giống nhau ở k ký tự đầu, tạo và in ra màn hình chuỗi S3 gồm k ký tự này (dùng hàm `strncpy`).



Bài tập

Bài 8: Nhập vào hai chuỗi ký tự str1 và str2.(chiều dài tối đa của mỗi chuỗi là 80 ký tự)

- Hãy cho biết số lần chuỗi str2 xuất hiện trong chuỗi str1.
- Nhập vào chuỗi str3, thay toàn bộ các chuỗi str2 trong str1 bằng str3.

Bài 9: Viết chương trình nhập vào một chuỗi, cho biết:

- Trong chuỗi có bao nhiêu chữ thường, chữ hoa, chữ số và ký hiệu khác (sử dụng các hàm isupper, islower, isdigit)
- Cho biết chuỗi có đối xứng hay không
- Tổng mã ASCII của tất cả ký tự của chuỗi.

Bài 10: Viết hàm standard(char s[]) bỏ toàn bộ khoảng trắng đầu chuỗi, cuối chuỗi và giữa 2 từ trong s chỉ còn 1 khoảng trắng.



Bài tập

Bài 11: Xóa tất cả các khoảng trắng của s.

Bài 12: Đếm xem có bao nhiêu từ trong s. Xuất các từ trên các dòng liên tiếp.

Bài 13: Cho 1 chuỗi (gồm nhiều từ). Hãy tìm một từ có chiều dài dài nhất và in ra màn hình.

Bài 14: Trích ra n ký tự đầu tiên/cuối cùng/bắt đầu tại vị trí pos.



Chúc các em học tốt!

