Bài 38: CÁC THAO TÁC TRÊN MẢNG KÝ TỰ (C-STYLE STRINGS)

Xem bài học trên website để ủng hộ Kteam: Các thao tác trên Mảng ký tự (C-style strings)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage How Kteam nhé!

Dẫn nhập

Ở bài học trước, mình đã chia sẻ cho các bạn bản chất và cách sử dụng MẢNG KÝ TỰ TRONG C++ (C-style strings).

Hôm nay, mình sẽ giới thiệu cho các bạn về **Các thao tác trên Mảng ký tự (C-style strings) trong C++**.

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về:

- KIỂU KÝ TỰ TRONG C++ (Character)
- MẢNG MỘT CHIỀU (Arrays)
- MẢNG KÝ TỰ TRONG C++ (C-style strings)

Trong bài ta sẽ cùng tìm hiểu các vấn đề:

- Một số thao tác với mảng ký tự (C-style strings)

Copyright © Howkteam.com

Một số thao tác với mảng ký tự (C-style strings)

Ngôn ngữ C++ cung cấp nhiều hàm để thao tác với **mảng ký tự (C-style strings)**, những hàm này được định nghĩa bên trong thư viện **<cstring>**.

Trong bài học này, mình sẽ giới thiệu một số hàm thường được sử dụng nhất trong C++.

Xem độ dài mảng ký tự (C-style strings)

Để biết được **độ dài** mảng ký tự **(không bao gồm ký tự null '\0')**, bạn có thể sử dụng hàm **strlen()**.

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
	char szTeam[20] = "Kteam"; // mảng có 20 phần tử (5 ký tự thường, 15 ký tự '\0')
	cout << "Team cua tui: " << szTeam << endl;
	cout << szTeam << " co " << strlen(szTeam) << " ky tu." << endl;
	cout << szTeam << " co " << sizeof(szTeam) << " phan tu trong mang." << endl;

	return 0;
}
```

Output:

Copyright © Howkteam.com

> **Lưu ý:** Hàm **strlen()** in ra số ký tự trước ký tự **'\0' null**, trong khi **sizeof()** trả về kích thước của **toàn bộ mảng**.

Chuyển mảng ký tự (C-style strings) sang chữ hoa và chữ thường

Để chuyển 1 chuỗi từ chữ thường sang chữ in hoa và ngược lại, bạn có thể sử dụng 2 hàm:

- **strlwr()**: chuyển chuỗi s thành **chuỗi thường** ('A' thành 'a', 'B' thành 'b', ..., 'Z' thành 'z').
- **strupr()**: chuyển chuỗi s thành **chuỗi IN hoa** ('a thành 'A', 'b' thành 'B', ..., 'z' thành 'Z').

Ví dụ:

```
#define _CRT_NONSTDC_NO_DEPRECATE
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
	char szString1[] = "Hello Howkteam.com!";
	char szString2[] = "Hello Howkteam.com!";

	cout << "s1: " << szString1 << endl;
```


Copyright © Howkteam.com

```
        cout << "s2: " << szString2 << endl;

        strlwr(szString1);
        strupr(szString2);
        cout << "s1: " << szString1 << endl;
        cout << "s2: " << szString2 << endl;

        return 0;
}
```

Output:

Một số compiler hiện đại thường cảnh báo về việc sử dụng hàm **strlwr()** và **strupr()**, và yêu cầu lập trình viên thêm dòng lệnh **#define _CRT_NONSTDC_NO_DEPRECATE** vào đầu chương trình để có thể sử dụng hàm **strlwr()** và **strupr()**.

Trong **C++ 11**, bạn có thể sử dụng 2 hàm **_strlwr_s()** và **_strupr_s()** để thay thế.

Sao chép mảng ký tự (C-style strings)

Để **sao chép** 1 chuỗi ký tự sang 1 chuỗi ký tự khác, bạn có thể sử dụng hàm **strcpy()**.

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cstring>
using namespace std;
```


Copyright © Howkteam.com

```
int main()
{
	char szSource[] = "Kteam";
	char szDest[20];

	// sao chép chuỗi szSource sang chuỗi szDest
	strcpy(szDest, szSource);
	cout << "Source: " << szSource << endl;
	cout << "Dest: " << szDest << endl;

	return 0;
}
```

Output:

> **Chú ý:** Khi sử dụng hàm này, chuỗi đích phải **đủ lớn để chứa được chuỗi nguồn**. Nếu không, vấn đề **tràn mảng** sẽ xảy ra.

Một số compiler hiện đại thường cảnh báo về việc sử dụng hàm **strcpy() là không an toàn**, và yêu cầu lập trình viên thêm dòng lệnh ***#define _CRT_SECURE_NO_WARNINGS*** vào đầu chương trình để có thể sử dụng hàm **strcpy()**.

Trong **C++ 11**, hàm **strcpy_s() được thay thế** cho hàm **strcpy()**, hàm này có thêm 1 tham số cho phép xác định độ dài của chuỗi đích. Nếu chuỗi đích không đủ lớn để chứa chuỗi nguồn, **compiler** sẽ ném ra 1 assert trong **debug mode**, và kết thúc chương trình.

```
#include <iostream>
```


Copyright © Howkteam.com

```
#include <cstring>
using namespace std;

int main()
{
	char szSource[] = "Howkteam.com";
	char szDest[5];

	// sao chép chuỗi szSource sang chuỗi szDest
	strcpy_s(szDest, 5, szSource); // 1 assert được ném ra trong debug mode
	cout << "Source: " << szSource << endl;
	cout << "Dest: " << szDest << endl;

	return 0;
}
```

Nối 2 mảng ký tự (C-style strings)

Để **nối** 1 chuỗi vào sau chuỗi khác, bạn có thể sử dụng hàm **strcat()**.

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
	char szSource[] = "Howkteam.com!";
	char szDest[100] = "Hello";

	// nối chuỗi
	strcat(szDest, " "); // "Hello "
	strcat(szDest, szSource); // "Hello Howkteam.com!"
	cout << "Dest: " << szDest << endl;

	return 0;
}
```


Copyright © Howkteam.com

Output:

> **Chú ý:** Khi sử dụng hàm **strcat(), chuỗi đích phải đủ lớn** để chứa được thêm chuỗi mới được nối nào. Nếu không, vấn đề **tràn mảng** sẽ xảy ra.

Trong **C++ 11**, hàm **strcat_s() được thay thế** cho hàm **strcat()**, hàm này có thêm 1 tham số cho phép xác định độ dài của chuỗi đích. Nếu chuỗi đích không đủ lớn để chứa thêm chuỗi nguồn, **compiler** sẽ ném ra 1 assert trong **debug mode**, và kết thúc chương trình.

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
	char szSource[] = "Howkteam.com!";
	char szDest[10] = "Hello";

	// nối chuỗi
	strcat_s(szDest, 10, " ");
	strcat_s(szDest, 10, szSource); // 1 assert được ném ra trong debug mode
	cout << "Dest: " << szDest << endl;

	return 0;
}
```


Copyright © Howkteam.com

So sánh 2 mảng ký tự (C-style strings)

Để **so sánh** hai chuỗi ký tự **s1** và **s2 (phân biệt hoa thường)**, bạn có thể sử dụng hàm **strcmp()**.

- Giá trị trả về **nhỏ hơn 0** nếu: chuỗi **s1 <** chuỗi **s2**
- Giá trị trả về **bằng 0** nếu: chuỗi **s1 ==** chuỗi **s2**
- Giá trị trả về **lớn hơn 0** nếu: chuỗi **s1 >** chuỗi **s2**

Ví dụ:

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
	char szString1[] = "howkteam.com!";
	char szString2[] = "Howkteam.com!";

	cout << "s1: " << szString1 << endl;
	cout << "s2: " << szString2 << endl;

	// so sánh 2 chuỗi
	int result = strcmp(szString1, szString2);
	if (result < 0)
		cout << "s1 < s2" << endl;
	else if (result > 0)
		cout << "s1 > s2" << endl;
	else
		cout << "s1 == s2" << endl;

	return 0;
}
```

Output:

Copyright © Howkteam.com

Tìm kiếm chuỗi trong chuỗi

Để tìm vị trí xuất hiện đầu tiên của một chuỗi (s2) trong một chuỗi khác (s1), bạn có thể sử dụng hàm strstr().

- Nếu **tìm thấy**: trả về **con trỏ đến vị trí xuất hiện đầu tiên** của chuỗi s2 trong chuỗi s1.
- Nếu **không tìm thấy**: trả về **NULL**.

Khái niệm **con trỏ** sẽ được nhắc tới trong bài CON TRỎ CƠ BẢN TRONG C++(Pointers).

Ví dụ:

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
	char szString1[] = "Hello Howkteam.com!";
	char szString2[] = "kteam";

	cout << "s1: " << szString1 << endl;
	cout << "s2: " << szString2 << endl;

	if (strstr(szString1, szString2) != NULL)
		cout << "Tim thay " << szString2 << " trong " << szString1 <<
endl;
	else
		cout << "Khong tim thay!" << endl;
```


Copyright © Howkteam.com

```
        return 0;
}
```

Output:

Kết luận

Qua bài học này, bạn đã biết được Các thao tác trên Mảng ký tự (C-style strings) trong C++. Còn rất nhiều thao tác khác trên mảng ký tự, trong phạm vi bài học không thể đề cập hết được, các bạn hãy tự mình tìm hiểu và bình luận bên dưới để chia sẻ cho mọi người nhé.

Trong bài tiếp theo, mình sẽ giới thiệu cho các bạn TỪ KHÓA AUTO TRONG C++ (The auto keyword).

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".

Copyright © Howkteam.com