

CSC10001

# HƯỚNG DẪN XỬ LÝ CHUỖI VỚI C++

FIT-HCMUS

## Contents

<b>1</b>	<b>Các thư viện giúp xử lý chuỗi thông dụng</b>	<b>2</b>
1.1	string . . . . .	2
1.2	<cstring> (string.h) . . . . .	2
<b>2</b>	<b>Một số hàm được sử dụng để xử lý chuỗi trong thư viện &lt;cstring&gt;</b>	<b>3</b>
2.1	strlen() . . . . .	3
2.2	strcpy() . . . . .	3
2.3	strcat() . . . . .	3
2.4	strcmp() . . . . .	4
2.5	strchr() . . . . .	4
2.6	strstr() . . . . .	4
2.7	strtok() . . . . .	5
2.8	to_string() và c_str() . . . . .	5
2.9	stoi() . . . . .	6
<b>3</b>	<b>Ví dụ mẫu</b>	<b>6</b>
<b>4</b>	<b>Một số lưu ý</b>	<b>8</b>

# 1 Các thư viện giúp xử lý chuỗi thông dụng

## 1.1 string

- Thư viện string hỗ trợ xử lý chuỗi có kiểu dữ liệu `std::string`.
- Thư viện này chỉ có ở ngôn ngữ C++.
- Ví dụ sử dụng:

```
#include <iostream>
#include <string> // length()

using namespace std;

int main ()
{
    string s = "";

    cout << "Enter a sentence: ";
    getline(cin, s);

    cout << "The content of s is: " << s << "\n";
    cout << "The size of s is " << s.length() << " bytes.\n";

    return 0;
}
```

- Sinh viên tự tìm hiểu thêm tại: [std::string C++](#)

## 1.2 <cstring> (string.h)

- Thư viện <cstring> hỗ trợ xử lý chuỗi có kiểu dữ liệu là mảng các ký tự (`char s[256]`).
- Thư viện này có ở cả ngôn ngữ C và C++.
- Ví dụ sử dụng:

```
#include <iostream>
#include <cstring> // strlen()

using namespace std;

int main()
{
    char s[256];

    cout << "Enter a sentence: ";
```

```

    gets(s);

    cout << "The content of sentence is: " << s << "\n";
    cout << "The size of sentence is: " << strlen(s) << " bytes.\n";

    return 0;
}

```

- Tham khảo thêm: `<cstring>` (`string.h`)

## 2 Một số hàm được sử dụng để xử lý chuỗi trong thư viện `<cstring>`

### 2.1 `strlen()`

- Dùng để tính độ dài chuỗi.
- Ví dụ:

```

char s[256] = "Sample string";

int len_s = strlen(s); // len_s = 13

```

### 2.2 `strcpy()`

- Dùng để sao chép nội dung của chuỗi nguồn đến chuỗi đích (kể cả ký tự kết thúc).
- Ví dụ:

```

char s_src[] = "Hello World"; // Chuoi nguồn
char s_des[256];              // Chuoi đích

strcpy(s_des, s_src);

// Sau khi thực hiện câu lệnh, s_des = "Hello World"

```

### 2.3 `strcat()`

- Dùng để ghép nối hai chuỗi.
- Ví dụ:

```

char str[256] = "Hello ";

strcat(str, "World");

// Sau khi thực hiện câu lệnh, str = "Hello World"

```

## 2.4 strcmp()

- Dùng để so sánh hai chuỗi `s1` và `s2`.
- Trả về:
  - + 0: nếu `s1` bằng (giống) `s2`.
  - + < 0: nếu ký tự đầu tiên không khớp trong chuỗi `s1` NHỎ HƠN chuỗi `s2`.
  - + > 0: nếu ký tự đầu tiên không khớp trong chuỗi `s1` LỚN HƠN chuỗi `s2`.
- Ví dụ:

```
char s1[256] = "hello";
char s2[256] = "hello";
char s3[256] = "hi";

int s1_vs_s2 = strcmp(s1, s2);
// s1_vs_s2 = 0 vì s1 giống s2

int s1_vs_s3 = strcmp(s1, s3);
// s1_vs_s3 < 0
// NHỎ HƠN 0 vì ký tự đầu tiên không khớp 'e' của s1 nhỏ hơn 'i' của s2
```

## 2.5 strchr()

- Dùng để tìm kiếm vị trí xuất hiện của một ký tự trong chuỗi.
- Ví dụ:

```
char str[] = "Hello world";
char * pch;
char find_char = 'u';

pch = strchr(str, find_char);

if (pch == NULL) // Không tìm thấy ký tự find_char trong chuỗi str
    cout << "Not found " << find_char << " in str!";
else // Tìm thấy ký tự find_char trong chuỗi str
    cout << "Found " << find_char << " at " << pch - str << "!";

// Lưu ý: vị trí ký tự find_char xuất hiện trong chuỗi str = pch - str
```

## 2.6 strstr()

- Dùng để tìm kiếm vị trí xuất hiện của một chuỗi trong một chuỗi.
- Ví dụ:

```
char str[] = "Hello World";
char* pch;
char find_str[] = "llo";

pch = strstr(str, find_str);

if (pch == NULL) // Không tìm thấy chuỗi find_str trong chuỗi str
    cout << "Not found " << find_str << " in str!\n";
else           // Tìm thấy chuỗi find_str trong chuỗi str
    cout << "Found " << find_str << " at " << pch - str << "!\n";

// Lưu ý: vị trí chuỗi find_str xuất hiện trong chuỗi str = pch - str
```

## 2.7 strtok()

- Dùng để tách chuỗi thành các token.
- Ví dụ:

```
// Ví dụ tách chuỗi str thành các token bởi các ký tự phân cách sau: khoảng trắng ( ), dấu
// phẩy (,), dấu chấm (.), dấu gạch ngang (-)

char str[] = "- This, a sample string.";
char * pch;

pch = strtok(str, " ,.-");
while (pch != NULL)
{
    std::cout << pch << "\n";
    pch = strtok(NULL, " ,.-");
}

/*
Ket qua xuất ra màn hình là:
This
a
sample
string
*/
```

## 2.8 to\_string() và c\_str()

- Đổi số thành chuỗi
- Thư viện: string

- Ví dụ:

```
char strDest[50];
int num = 50;
strcpy(strDest, to_string(num).c_str()); //strDest: "50"
```

## 2.9 stoi()

- Đổi chuỗi số thành số
- Thư viện: `string`
- Ví dụ:

```
char strDest[50] = "50";
string myString(strDest); // convert char array to string
int num = stoi(myString); //num = 50
```

## 3 Ví dụ mẫu

Đề bài: Nhập vào chuỗi chứa họ và tên của một người bất kỳ. In ra họ và tên người đó theo định dạng:  
Tên Họ.

Ví dụ:

- Input: "Nguyen Van Ty"
- Output: "Ty Nguyen"

Bài giải:

```
// Cach 1:

#include <iostream>
#include <cstring>

using namespace std;

int main ()
{
    char name[256];
    char split_name[10][256]; // Mang cac token duoc tach ra tu name
    int i = 0;    // Chi so cho mang split_name
    char * pch;

    cout << "Enter a name: ";
    gets(name);
```

```

    pch = strtok (name, " ");
    strcpy(split_name[i++], pch);

    while (pch != NULL)
    {
        strcpy(split_name[i++], pch);
        pch = strtok (NULL, " ");
    }

    cout << split_name[i-1] << " " << split_name[0];

    return 0;
}

```

```

// Cach 2: (Nang cao)

#include <iostream>
#include <cstring>

using namespace std;

int main ()
{
    char name[256], family_name[256], given_name[256];
    char* first_space;
    char* last_space;

    cout << "Enter a name: ";
    gets(name);

    first_space = strchr(name, ' '); // Tim dau cach dau tien de lay ho
    int len_family_name = first_space - name;
    strncpy(family_name, name, len_family_name); // Copy ho tu name vao family_name
    family_name[len_family_name] = '\0'; // Gan ky tu ket thuc cho ho

    last_space = strrchr(name, ' '); // Tim dau cach cuoi cung de lay ten
    strcpy(given_name, last_space + 1); // Copy ten tu name vao given_name

    cout << given_name << " " << family_name;

    return 0;
}

```

## 4 Một số lưu ý

1. Khi khai báo một chuỗi ở kiểu dữ liệu `std::string` → sử dụng thư viện `<string>` (`#include <string>`).
2. Khi khai báo một chuỗi ở kiểu dữ liệu là mảng các ký tự (`char s[256]`) → sử dụng thư viện `<cstring>` (`#include <cstring>`).
3. Khi thực hiện các thao tác trên chuỗi (mảng các ký tự (`char s[256]`)) phải bảo đảm có ký tự kết thúc `'\0'`.
4. (\*) Khi nhận được chuỗi đầu vào `s` có kiểu dữ liệu `std::string`, ta hoàn toàn có thể sử dụng thư viện `<cstring>` để xử lý bằng cách sử dụng hàm `c_str()` để "biến `s` từ `std::string` thành `s` là mảng ký tự". Tham khảo thêm tại: `c_str C++`.