**BÁO CÁO**

**ĐỀ TÀI SEMINAR**

***GVHD*** *: Trương Toàn Thịnh*

*Lớp 15CTT1*

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Bộ môn Phương Pháp Lập Trình Hướng Đối Tượng

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

**MỤC LỤC**

Table of Contents

[I. Thông tin nhóm 4](#_Toc467105595)

[II. Giới thiệu nội dung đề tài : 5](#_Toc467105596)

[III. Thư viện chuỗi (*Strings* libraries): 6](#_Toc467105597)

[1. Tìm hiểu std::basic\_string: 6](#_Toc467105598)

[a. Khái niệm: 6](#_Toc467105599)

[b. Đặc điểm: 7](#_Toc467105600)

[c. Một số loại thông dụng: 7](#_Toc467105601)

[d. Hàm thành phần: 7](#_Toc467105602)

[e. Giá trị đặt biệt: 10](#_Toc467105603)

[f. Hàm ngoài: 10](#_Toc467105604)

[g. Hàm nhập xuất: 10](#_Toc467105605)

[h. Hàm chuyển đổi: 11](#_Toc467105606)

[i. Literals: 11](#_Toc467105607)

[2. Tìm hiểu std::basic\_string\_view: 12](#_Toc467105608)

[a. Khái niệm: 12](#_Toc467105609)

[b. Đặc điểm: 12](#_Toc467105610)

[c. Một số loại thông dụng: 12](#_Toc467105611)

[d. Hàm thành phần: 13](#_Toc467105612)

[3. Tìm hiểu null-terminated strings: 16](#_Toc467105613)

[3.1. Null-terminated byte strings: 16](#_Toc467105614)

[3.2. Null-terminated multibyte strings: 19](#_Toc467105615)

[3.3. Null-terminated wide byte strings: 20](#_Toc467105616)

[IV. Thư viện nhập xuất (*IOstreams* libraries): 23](#_Toc467105617)

[1. Khái niệm và các thành phần chính: 23](#_Toc467105618)

[2. Input/Output với file trong C++: 24](#_Toc467105619)

[3. Stream cho chuỗi: 25](#_Toc467105620)

[4. Các lớp thành phần: 25](#_Toc467105621)

[5. Các đối tượng: 25](#_Toc467105622)

[6. Loại: 26](#_Toc467105623)

[7. Các thao tác: 26](#_Toc467105624)

[V. Các thuật toán (*Algorithms* STL): 27](#_Toc467105625)

[1. Khái niệm: 27](#_Toc467105626)

[2. Các thao tác không làm thay đổi dữ liệu: 27](#_Toc467105627)

[3. Các thao tác làm thay đổi dữ liệu: 28](#_Toc467105628)

[4. Thao tác phân vùng: 28](#_Toc467105629)

[5. Thao tác sắp xếp: 29](#_Toc467105630)

[6. Thao toán tìm kiếm nhị phân: 29](#_Toc467105631)

[7. Thao tác thiết đặt: 29](#_Toc467105632)

[8. Thao tác vun đống: 29](#_Toc467105633)

[9. Thao tác min/max: 30](#_Toc467105634)

[10. Thao tác số học: 30](#_Toc467105635)

[11. Thao tác trên bộ nhớ chưa được định hình: 30](#_Toc467105636)

[12. Thư viện C: 31](#_Toc467105637)

[VI. Thông tin bổ sung: 32](#_Toc467105638)

[ **Các nguồn tài liệu tham khảo chính:** 32](#_Toc467105639)

1. **Thông tin nhóm**

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ tên** | **Email** | **Điện thoại** |
| 1412590 | Phan Quốc Trung | [trungpq999@gmail.com](mailto:trungpq999@gmail.com) | 0966 187 219 |

1. **Giới thiệu nội dung đề tài :**

Standard Template Library (STL) là một thư viện chuẩn của C++ được hầu hết các môi trường lập trình hỗ trợ. Nó là tập hợp các hàm, lớp, đối tương và các mẫu dựa trên ngôn ngữ C++ nhằm cung cấp các chức năng và công cụ hỗ trợ lập trình viên trong quá trình xây dựng ứng dụng. Về cơ bản STL bao gồm:

* Thư viện C (*C* library).
* Thư viện chuỗi và nhập/xuất (*Strings* and *IOstream* libraries).
* Các cấu trúc (*Containers* STL).
* Các thuật toán (*Algorithms* STL).

Trong số 4 thành phần cơ bản của thư viện STL trên, nhóm sẽ đi sâu vào nghiêm cứu và báo cáo về 3 thành phần cơ bản là:

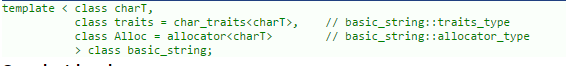
1. Thư viện chuỗi(*Strings libraries)*
2. Thư viện nhập/xuất (*IOstream* libraries).
3. Các thuật toán (*Algorithms* STL).
4. **Thư viện chuỗi (*Strings* libraries):**

Thư viện chuỗi được sử dụng để cung cấp các thao tác cho 3 loại chuỗi phổ biến:

* std::basic\_string : một template được thiết kế để thao tác trên bất cứ kiểu dữ liệu ký tự nào.
* std::basic\_string\_view(c++17) : một đối tượng không chứa bất dữ liệu ký tự chỉ có thao tác đọc đối với chuỗi ký tự hiện hành.
* Null-terminated strings: là một mảng các ký tự và được kết thúc bằng một ký tự *null*.
  1. Tìm hiểu std::basic\_string:

1. Khái niệm:

* Class template class std::basic\_string như là một loại khái quát hóa của lớp string cho tất cả các loại dữ liệu.
* Class template std::basic\_string lưu trữ và cung cấp các thao tác đối với các chuỗi ký tự-giống như một đối tượng. Các thao tác tạo chuỗi, hủy chuỗi và các thao tác đặc thù khác đều được hỗ trợ bởi class template std::basic\_string.
* Class template std::basic\_string



Các tham số:

+ charT : kiểu ký tự.

+ traits : định nghĩa thuộc tính của các đối tượng được sử dụng bởi basic\_string.

+ Alloc : loại đối tượng cấp phát sử dụng để xác địng mô hình phần bổ lưu trữ. Mặc định lớp mẫu cấp phát được sử dụng, là đối tượng xác định mô hình cấp phát bộ nhớ là đơn giản nhất và là giá trị độc lập

1. Đặc điểm:

* Std::basic\_string không phụ thuộc vào loại dữ liệu của ký tự và tính chất hoạt động của dữ liệu đó. Phép toán định nghĩa được cung cấp tham số template Traits – một hàm trong std::char\_traits.
* Các phần tử của một basic\_string được lữu trữ liên tục với nhau, nếu như ta có một basic\_string s.

1. Một số loại thông dụng:

* Một số loại thường dùng được cung cấp bởi template class std::basic\_string:

|  |  |  |
| --- | --- | --- |
| Type | Class | Definition |
| std::string | string | std::basic\_string<char\_t> |
| std::wstring | wide string | std::basic\_string<wchar\_t> |
| std::u16string | string of 16-bit characters | std::basic\_string<char16\_t> |
| std::u32string | wstring of 32-bit characters | std::basic\_string<char32\_t> |

1. Hàm thành phần:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| (constructor) | Tạo một basic\_string | Public |
| (destructor) | Hủy string, giải phóng vùng nhớ | Public |
| operator= | Gán giá trị vào chuỗi | Public |
| assign | Gán các ký tự vào chuỗi | Public |
| get\_allocator | Trả về cấp phát liên quan | Public |

* Hàm truy cập:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| at | Xuất ra ký tự tại vị trí chỉ định | Public |
| [operator[]](http://en.cppreference.com/w/cpp/string/basic_string/operator_at) | Xuất ra ký tự tại vị trí chỉ định | Public |
| [front](http://en.cppreference.com/w/cpp/string/basic_string/front) | Trả ra địa chỉ phần tử đầu tiên | Public |
| [back](http://en.cppreference.com/w/cpp/string/basic_string/back) | Trả ra địa chỉ phần tử đầu tiên | Public |
| [data](http://en.cppreference.com/w/cpp/string/basic_string/data) | Cung cấp một con trỏ đến mảng phục vụ việc lưu trữ ký tự của chuỗi, | Public |
| [c\_str](http://en.cppreference.com/w/cpp/string/basic_string/c_str) | Trả về một mảng ký tự của string theo định dạng C | Public |
| [operator basic\_string\_view](http://en.cppreference.com/w/cpp/string/basic_string/operator_basic_string_view) | Trả về một chuỗi chỉ để xem từ chuỗi ban đầu | Public |

* Iterators:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| [begin](http://en.cppreference.com/w/cpp/string/basic_string/begin)  cbegin | Trả về một iterator đến ký tự đầu tiên của chuỗi | Public |
| end  cend | Trả về một iterator đến ký tự cuối cùng của chuỗi | Public |
| [rbegin](http://en.cppreference.com/w/cpp/string/basic_string/rbegin)  crbegin | Chuyển iterator kết thúc thành iterator bắt đầu | Public |
| [rend](http://en.cppreference.com/w/cpp/string/basic_string/rend)  crend | Chuyển iterator bắt đầu thành iterator kết thúc | Public |

* Hàm định dạng lưu trữ:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| [empty](http://en.cppreference.com/w/cpp/string/basic_string/empty) | Kiểm tra chuỗi có rỗng hay không | Public |
| [size](http://en.cppreference.com/w/cpp/string/basic_string/size)  length | Trả về số ký tự trong chuỗi | Public |
| [max\_size](http://en.cppreference.com/w/cpp/string/basic_string/max_size) | Trả về số ký tự tối đa của chuỗi | Public |
| reserve | Đảo ngược chuỗi | Public |
| [capacity](http://en.cppreference.com/w/cpp/string/basic_string/capacity) | Trả về số ký tự của chuỗi chiếm dụng bộ nhớ | Public |
| [shrink\_to\_fit](http://en.cppreference.com/w/cpp/string/basic_string/shrink_to_fit) | Giải phóng vùng nhớ không sử dụng | Public |

* Hàm thao tác phần tử:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| clear | Xóa nội dung | Public |
| [insert](http://en.cppreference.com/w/cpp/string/basic_string/insert) | Chèn nội dung | Public |
| [erase](http://en.cppreference.com/w/cpp/string/basic_string/erase) | Xóa ký tự | Public |
| [push\_back](http://en.cppreference.com/w/cpp/string/basic_string/push_back) | Thêm vào cuối chuỗi | Public |
| [pop\_back](http://en.cppreference.com/w/cpp/string/basic_string/pop_back) | Lấy ra và xóa ký tự cuối cùng trong chuỗi | Public |
| [append](http://en.cppreference.com/w/cpp/string/basic_string/append) | Nối ký tự vào cuối chuỗi | Public |
| [operator+=](http://en.cppreference.com/w/cpp/string/basic_string/operator%2B%3D) | Nối ký tự vào cuối chuỗi | Public |
| [compare](http://en.cppreference.com/w/cpp/string/basic_string/compare) | So sánh 2 chuỗi | Public |
| [replace](http://en.cppreference.com/w/cpp/string/basic_string/replace) | Thay thế một phần chỉ định của chuỗi | Public |
| substr | Trả ra chuỗi con | Public |
| [copy](http://en.cppreference.com/w/cpp/string/basic_string/copy) | Copy ký tự | Public |
| resize | Thay đổi số lượng ký tự lưu trữ | Public |
| [swap](http://en.cppreference.com/w/cpp/string/basic_string/swap) | Hoán đổi nội dung | Public |

* Hàm tìm kiếm:

|  |  |  |
| --- | --- | --- |
| Tên hàm Tên hàm | Chức năng | Mức độ truy cập |
| [find](http://en.cppreference.com/w/cpp/string/basic_string/find) | Tìm ký tự trong chuỗi | Public |
| [rfind](http://en.cppreference.com/w/cpp/string/basic_string/rfind) | Tìm vị trí xuất hiện cuối cùng của chuỗi con trong chuỗi | Public |
| find\_first\_of | Tìm vị trí xuất đầu tiên của chuỗi con trong chuỗi | Public |
| [find\_first\_not\_of](http://en.cppreference.com/w/cpp/string/basic_string/find_first_not_of) | Tìm vị trí không xuất hiện đầu tiên của chuỗi con trong chuỗi | Public |
| [find\_last\_of](http://en.cppreference.com/w/cpp/string/basic_string/find_last_of) | Tìm vị trí xuất hiện cuối cùng của chuỗi con trong chuỗi | Public |
| [find\_last\_not\_of](http://en.cppreference.com/w/cpp/string/basic_string/find_last_not_of) | Tìm vị trí không xuất hiện cuối cùng của chuỗi con trong chuỗi | Public |

1. Giá trị đặt biệt:

[npos](http://en.cppreference.com/w/cpp/string/basic_string/npos) : một giá trị đặc biệt, giá trị chính xác phụ thuộc vào loại nội dung.

1. Hàm ngoài:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| operator+ | Cộng 2 chuỗi |  |
| operator==  operator!=  operator<  operator>  operator<=  operator>= | Các phép so sánh 2 chuỗi |  |

1. Hàm nhập xuất:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| [operator<<](http://en.cppreference.com/w/cpp/string/basic_string/operator_ltltgtgt) | Xuất chuỗi |  |
| operator>> | Nhập chuỗi |  |
| getline | Đọc dữ liệu từ I/O stream vào string |  |

1. Hàm chuyển đổi:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| [stoi](http://en.cppreference.com/w/cpp/string/basic_string/stol" \o "cpp/string/basic string/stol)  [stol](http://en.cppreference.com/w/cpp/string/basic_string/stol" \o "cpp/string/basic string/stol)  [stoll](http://en.cppreference.com/w/cpp/string/basic_string/stol" \o "cpp/string/basic string/stol) | Chuyển chuỗi thành số nguyên có dấu |  |
| [stoul](http://en.cppreference.com/w/cpp/string/basic_string/stoul" \o "cpp/string/basic string/stoul)  [stoull](http://en.cppreference.com/w/cpp/string/basic_string/stoul" \o "cpp/string/basic string/stoul) | Chuyển chuỗi thành số nguyên không dấu |  |
| [stof](http://en.cppreference.com/w/cpp/string/basic_string/stof" \o "cpp/string/basic string/stof)  [stod](http://en.cppreference.com/w/cpp/string/basic_string/stof" \o "cpp/string/basic string/stof)  [stold](http://en.cppreference.com/w/cpp/string/basic_string/stof" \o "cpp/string/basic string/stof) | Chuyển chuỗi thành số thực dấu chấm động |  |
| [to\_string](http://en.cppreference.com/w/cpp/string/basic_string/to_string) | Chuyển số thực dấu chấu động thành string |  |
| [to\_wstring](http://en.cppreference.com/w/cpp/string/basic_string/to_wstring) | Chuyển số thực dấu chấu động thành wstring |  |

1. Literals:

[operator""s](http://en.cppreference.com/w/cpp/string/basic_string/operator%22%22s) : chuyển mảng ký tự thành basic\_string

* 1. Tìm hiểu std::basic\_string\_view:
     + 1. Khái niệm:
* Template class std::basic\_stirng\_view cung cấp một đối có thể tham chiếu đến một chuỗi ký tự cố định giống như một đối tượng với phần từ đầu tiên của đối tượng đó ở vị trí số 0 của chuỗi được tham chiếu.
* Đối tượng tham chiếu này chỉ giữ 2 giá trị là: địa chỉ phần tử đầu tiên của chuỗi và kích thước chuỗi.
* Class template std::basic\_string\_view:



Các tham số:

+ CharT: loại ký tự.

+ Traits: lớp chỉ định các thao tác đối với kiểu dữ liễu charT.

* + - 1. Đặc điểm:
* basic\_string\_view là một công cụ để hỗ trợ cho việc thao tác trong template class basic\_string đơn giản và nhanh chóng hơn.
  + - 1. Một số loại thông dụng:

Một số loại thông dụng được cung cấp bởi template class std::basic\_string\_view:

|  |  |
| --- | --- |
| Type | Definition |
| std::string\_view | std::basic\_string\_view <char\_t> |
| std::wstring\_view | std::basic\_string\_view <wchar\_t> |
| std::u16string\_view | std::basic\_string\_view <char16\_t> |
| std::u32string\_view | std::basic\_string\_view <char32\_t> |

* + - 1. Hàm thành phần:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| (constructor) | Khởi tạo đối tượng string\_view | Public |
| Operator= | Gán bằng giá trị khác | Public |

* Iterators:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| [begin](http://en.cppreference.com/w/cpp/string/basic_string/begin)  cbegin | Trả về một iterator đến ký tự đầu tiên của chuỗi | Public |
| end  cend | Trả về một iterator đến ký tự cuối cùng của chuỗi | Public |
| [rbegin](http://en.cppreference.com/w/cpp/string/basic_string/rbegin)  crbegin | trả về một iterator ngược lại để bắt đầu | Public |
| [rend](http://en.cppreference.com/w/cpp/string/basic_string/rend)  crend | trả về một iterator ngược lại để kết thúc | Public |

* Hàm tham tác phần tử:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| at | Truy cập ký tự chỉ định để kiểm tra. | Public |
| [operator[]](http://en.cppreference.com/w/cpp/string/basic_string/operator_at) | Truy cập ký tự chỉ định | Public |
| [front](http://en.cppreference.com/w/cpp/string/basic_string/front) | Truy cập ký tự đầu tiên của chuỗi | Public |
| [back](http://en.cppreference.com/w/cpp/string/basic_string/back) | Truy cập ký tự cuối cùng của chuỗi | Public |
| [data](http://en.cppreference.com/w/cpp/string/basic_string/data) | Trả về con trỏ của phần tử đầu tiên | Public |

* Hàm định dạng lưu trữ:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| Size  length | Lấy số lượng ký tự | Public |
| max\_size | Lấy số ký tự tối đa | Public |
| [empty](http://en.cppreference.com/w/cpp/string/basic_string_view/empty) | Kiểm tra có phải là rông | Public |

* Hàm điều chỉnh:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| [remove\_prefix](http://en.cppreference.com/w/cpp/string/basic_string_view/remove_prefix) | Di chuyển con trỏ từ phía phần tử đầu tiên về phần tử sau. | Public |
| [remove\_suffix](http://en.cppreference.com/w/cpp/string/basic_string_view/remove_suffix) | Di chuyển con trỏ từ phía phần tử cuối cùng về phần tử trước | Public |
| [swap](http://en.cppreference.com/w/cpp/string/basic_string_view/swap) | Hoán đổi nội dung | Public |

* Hàm thao tác:

|  |  |  |
| --- | --- | --- |
| Tên hàm Tên hàm | Chức năng | Mức độ truy cập |
| [copy](http://en.cppreference.com/w/cpp/string/basic_string_view/copy) | Sao chép các ký tự | Public |
| [substr](http://en.cppreference.com/w/cpp/string/basic_string_view/substr) | Lấy chuỗi con | Public |
| [compare](http://en.cppreference.com/w/cpp/string/basic_string_view/compare) | So sánh 2 chuỗi | Public |
| [find](http://en.cppreference.com/w/cpp/string/basic_string/find) | Tìm ký tự trong chuỗi | Public |
| [rfind](http://en.cppreference.com/w/cpp/string/basic_string/rfind) | Tìm vị trí xuất hiện cuối cùng của chuỗi con trong chuỗi | Public |
| find\_first\_of | Tìm vị trí xuất đầu tiên của chuỗi con trong chuỗi | Public |
| [find\_first\_not\_of](http://en.cppreference.com/w/cpp/string/basic_string/find_first_not_of) | Tìm vị trí không xuất hiện đầu tiên của chuỗi con trong chuỗi | Public |
| [find\_last\_of](http://en.cppreference.com/w/cpp/string/basic_string/find_last_of) | Tìm vị trí xuất hiện cuối cùng của chuỗi con trong chuỗi | Public |
| [find\_last\_not\_of](http://en.cppreference.com/w/cpp/string/basic_string/find_last_not_of) | Tìm vị trí không xuất hiện cuối cùng của chuỗi con trong chuỗi | Public |

* Giá trị cố định:

[npos](http://en.cppreference.com/w/cpp/string/basic_string/npos) : một giá trị đặc biệt, giá trị chính xác phụ thuộc vào loại nội dung.

* Hàm ngoài:

|  |  |  |
| --- | --- | --- |
| Tên hàm | Chức năng | Mức độ truy cập |
| operator==  operator!=  operator<  operator>  operator<=  operator>= | Các phép so sánh 2 chuỗi |  |

* Hàm nhập xuất:

[operator<<](http://en.cppreference.com/w/cpp/string/basic_string_view/operator_ltlt) : xuất ra đối tượng string\_view.

* 1. Tìm hiểu null-terminated strings:

Null-terminated strings là một chuỗi các ký tự và được kết thúc bằng một ký tự null đăc biệt. C++ cung cấp các hàm tạo, kiểm duyệt, và chỉnh sửa null-terminated strings.

Có 3 loại null-terminated strings:

* Null-terminated byte strings.
* Null-terminated multibyte strings.
* Null-terminated wide byte strings.
  1. Null-terminated byte strings:
     + 1. Khái niệm:

Là một dãy các byte với giá trị khác 0 và được kết thức bởi byte có giá trị bằng 0.

* + - 1. Hàm phân loại ký tự:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [isalnum](http://en.cppreference.com/w/cpp/string/byte/isalnum) | Kiểm tra có là chữ số hoặc chữ, nếu đúng thì trả ra số khác 0 |
| [isalpha](http://en.cppreference.com/w/cpp/string/byte/isalpha) | Kiểm tra có là chữ cái, nếu đúng thì trả ra số khác 0 |
| [islower](http://en.cppreference.com/w/cpp/string/byte/islower) | Kiểm tra có là ký tự viết thường, nếu đúng thì trả ra số khác 0 |
| isupper | Kiểm tra có là ký tự viết hoa, nếu đúng thì trả ra số khác 0 |
| isdigit | Kiểm tra có là số, nếu đúng thì trả ra số khác 0 |
| [isxdigit](http://en.cppreference.com/w/cpp/string/byte/isxdigit) | Kiểm tra có là ký tự hexa , nếu đúng thì trả ra giá trị hexa của ký tự |
| iscntrl | Kiểm tra có là control character, nếu đúng thì trả ra số khác 0 |
| isgraph | Kiểm tra có là ký tự đồ họa, nếu đúng thì trả ra số khác 0 |
| [isspace](http://en.cppreference.com/w/cpp/string/byte/isspace) | Kiểm tra có là khoảng trắng, nếu đúng thì trả ra số khác 0 |
| [isblank](http://en.cppreference.com/w/cpp/string/byte/isblank) | Kiểm tra có là ký tự rỗng, nếu đúng thì trả ra số khác 0 |
| isprint | Kiểm tra có là ký tự in, nếu đúng thì trả ra số khác 0 |
| ispunct | Kiểm tra có là ký tự chấm, nếu đúng thì trả ra số khác 0 |

* + - 1. Hàm thao tác với ký tự:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [tolower](http://en.cppreference.com/w/cpp/string/byte/tolower) | Chuyển thành ký tự viết thường |
| toupper | Chuyển thành ký tự viết hoa |

* + - 1. Hàm chuyển đổi sang số:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [atof](http://en.cppreference.com/w/cpp/string/byte/atof) | Chuyển thành giá trị số thực dấu chấm động |
| [atoi](http://en.cppreference.com/w/cpp/string/byte/atoi" \o "cpp/string/byte/atoi)  [atol](http://en.cppreference.com/w/cpp/string/byte/atoi" \o "cpp/string/byte/atoi)  [atoll](http://en.cppreference.com/w/cpp/string/byte/atoi" \o "cpp/string/byte/atoi) | Chuyển thành giá trị số nguyên |
| [strtol](http://en.cppreference.com/w/cpp/string/byte/strtol" \o "cpp/string/byte/strtol)  [strtoll](http://en.cppreference.com/w/cpp/string/byte/strtol" \o "cpp/string/byte/strtol) | Chuyển thành giá trị số nguyên |
| strtoul  strtoull | Chuyển thành giá trị số nguyên dương |
| [strtof](http://en.cppreference.com/w/cpp/string/byte/strtof" \o "cpp/string/byte/strtof)  [strtod](http://en.cppreference.com/w/cpp/string/byte/strtof" \o "cpp/string/byte/strtof)  [strtold](http://en.cppreference.com/w/cpp/string/byte/strtof" \o "cpp/string/byte/strtof) | Chuyển thành giá trị số thực dấu chấm động |
| [strtoimax](http://en.cppreference.com/w/cpp/string/byte/strtoimax" \o "cpp/string/byte/strtoimax)  [strtoumax](http://en.cppreference.com/w/cpp/string/byte/strtoimax" \o "cpp/string/byte/strtoimax) | Chuyển thành  [std::intmax\_t](http://en.cppreference.com/w/cpp/types/integer) hoặc  [std::intmax\_t](http://en.cppreference.com/w/cpp/types/integer) |

* + - 1. Hàm thao tác với chuỗi:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [strcpy](http://en.cppreference.com/w/cpp/string/byte/strcpy) | Sao chép chuỗi |
| [strncpy](http://en.cppreference.com/w/cpp/string/byte/strncpy) | Sao chép số ký tự nhất định |
| [strcat](http://en.cppreference.com/w/cpp/string/byte/strcat) | Nối 2 chuỗi |
| strncat | Nối 2 chuỗi với số ký tự nhất định |
| [strxfrm](http://en.cppreference.com/w/cpp/string/byte/strxfrm) | So sánh 2 chuỗi |

* + - 1. Hàm truy vấn:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| strlen | Trả ra độ dài chuỗi |
| [strcmp](http://en.cppreference.com/w/cpp/string/byte/strcmp) | So sánh 2 chuỗi, trả ra 0 nếu bằng nhau |
| [strncmp](http://en.cppreference.com/w/cpp/string/byte/strncmp) | So sánh 2 chuỗi với số ký tự nhất định, trả ra 0 nếu bằng nhau |
| [strcoll](http://en.cppreference.com/w/cpp/string/byte/strcoll) | So sánh 2 chuỗi theo vị trí con trỏ lưu trữ hiện tại, trả ra 0 nếu bằng nhau |
| [strchr](http://en.cppreference.com/w/cpp/string/byte/strchr) | Trả về con trỏ tại vị trí có sự xuất hiện đầu tiên của một ký tự được chỉ định trước. |
| strrchr | Trả về con trỏ tại vị trí có sự xuất hiện cuối cùng của một ký tự được chỉ định trước. |
| strspn | Trả về độ dài của phần đầu tiên của str1 mà chỉ bao gồm các ký tự mà là một phần của str2. |
| strcspn | Trả về độ dài của phần đầu tiên của str1 mà chỉ bao gồm các ký tự mà không là một phần của str2. |
| strpbrk | Trả ra con trỏ tại vị trí đầu tiên của bất kỳ ký tự của str2 xuất hiện trong str1 |
| strstr | Tìm chuỗi con đầu tiên của chuỗi str2 xuất hiện trong str1 |
| strtok | Tách chuỗi dựa theo các ký tự quy định |

* + - 1. Hàm thao tác với mảng ký tự:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [memchr](http://en.cppreference.com/w/cpp/string/byte/memchr) | Tìm mảng đầu tiên xuất hiện ký tự cần tìm |
| [memcmp](http://en.cppreference.com/w/cpp/string/byte/memcmp) | So sánh 2 mảng, trả ra 1 nếu str1 > str2, trả ra 0 nếu bằng nhau, trả ra -1 str1<str2 |
| [memset](http://en.cppreference.com/w/cpp/string/byte/memset) | Gán 1 ký tự cho tất cả vùng nhớ |
| memcpy | Sao chép vùng nhớ |
| memmove | Di chuyển vùng nhớ |

* + - 1. Hàm khác:

[strerror](http://en.cppreference.com/w/cpp/string/byte/strerror) : trả về một phiên bản văn bản của một mã lỗi được đưa ra.

* 1. Null-terminated multibyte strings:

1. Khái niệm:

Là một dãy các byte khác không và được kết thúc bởi byte có giá trị là 0, mỗi ký tự trong chuỗi có thể được lưu trữ bởi nhiều hơn 1 byte.

1. Hàm chuyển đổi multibyte/wide character:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [mblen](http://en.cppreference.com/w/cpp/string/multibyte/mblen) | Trả về số byte dùng để biểu diễn ký tự multibyte character |
| [mbtowc](http://en.cppreference.com/w/cpp/string/multibyte/mbtowc) | Chuyển multibyte character thành wide character |
| [wctomb](http://en.cppreference.com/w/cpp/string/multibyte/wctomb) | Chuyển một wide character thành dạng multibyte của nó |
| [mbstowcs](http://en.cppreference.com/w/cpp/string/multibyte/mbstowcs) | Chuyển đổi một multibyte thành wide string |
| [wcstombs](http://en.cppreference.com/w/cpp/string/multibyte/wcstombs) | Chuyển một wide string thành một dãy multibyte |
| [mbsinit](http://en.cppreference.com/w/cpp/string/multibyte/mbsinit) | 0 nếu ps không phải là một con trỏ null và không đại diện cho trạng thái chuyển đổi ban đầu, giá trị không không khác. |
| [btowc](http://en.cppreference.com/w/cpp/string/multibyte/btowc) | Nếu c là một ký tự đơn byte có giá trị trong trạng thái thay đổi ban đầu của một chuỗi nhiều byte, hàm trả về đại diện của mình như một wchar\_t (loại đúc đến một giá trị kiểu wint\_t).  Nếu c là EOF, hoặc nếu c không phải là một đại diện byte đơn hợp lệ, hàm trả về WEOF. |
| [wctob](http://en.cppreference.com/w/cpp/string/multibyte/wctob) | Trả về biểu diễn single-byte của wide character wc nếu (và chỉ nếu) wc tương ứng với một ký tự nhiều byte với chiều dài của một byte duy nhất trong tình trạng ban đầu của một chuỗi nhiều byte.  Nếu không, nó sẽ trả về EOF. |
| [mbrlen](http://en.cppreference.com/w/cpp/string/multibyte/mbrlen) | Trả về số byte của multibyte character. |
| [mbrtowc](http://en.cppreference.com/w/cpp/string/multibyte/mbrtowc) | Chuyển multibyte character thành wide character |
| [wcrtomb](http://en.cppreference.com/w/cpp/string/multibyte/wcrtomb) | Chuyển đổi wide string thành dạng multibyte character. |
| [mbsrtowcs](http://en.cppreference.com/w/cpp/string/multibyte/mbsrtowcs) | Chuyển một chuôi multibyte character thành wide character |
| [wcsrtombs](http://en.cppreference.com/w/cpp/string/multibyte/wcsrtombs) | Chuyển wide character thành multibyte character |
| [mbrtoc16](http://en.cppreference.com/w/cpp/string/multibyte/mbrtoc16) | Chuyển chuỗi ký tự multibyte thành dạng ký tự 16bit |
| [c16rtomb](http://en.cppreference.com/w/cpp/string/multibyte/c16rtomb) | Chuyển đổi 16bite wide chararacter thành chuỗi multibyte |
| [mbrtoc32](http://en.cppreference.com/w/cpp/string/multibyte/mbrtoc32) | Tạo ra 32-bit wide character tiếp theo dựa vào multibyte character |
| [c32rtomb](http://en.cppreference.com/w/cpp/string/multibyte/c32rtomb) | Chuyển đổi 32bit wide chararacter thành chuỗi multibyte |

1. Hàm kiểu mẫu:

[mbstate\_t](http://en.cppreference.com/w/cpp/string/multibyte/mbstate_t) : chuyển đổi thông tin trạng thái cần thiết để lặp chuỗi ký tự nhiều byte

1. Macros:

|  |  |
| --- | --- |
| Tên | Giá Trị |
| MB\_LEN\_MAX | Số byte tối đa của trong một multibyte character |
| MB\_CUR\_MAX | Số byte tối đa của trong một multibyte character tại vùng lưu trữ hiện tại |
| \_\_STDC\_UTF\_16\_\_ | Biểu diễn mà UTF-16 sử dụng mbrtoc16 và c16rtomb |
| \_\_STDC\_UTF\_32\_\_ | Biểu diễn mà UTF-32 sử dụng mbrtoc32 và c32rtomb |

* 1. Null-terminated wide byte strings:

1. Khái niệm:

Là một chuỗi rộng(chuỗi với các phần tử được lưu trữ với nhiều hơn 1 byte) với trình tự hợp lệ và kết thức bằng ký tự null.

1. Hàm phân loại ký tự:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [isalnum](http://en.cppreference.com/w/cpp/string/byte/isalnum) | Kiểm tra có là chữ số |
| [isalpha](http://en.cppreference.com/w/cpp/string/byte/isalpha) | Kiểm tra có là chữ cái |
| [islower](http://en.cppreference.com/w/cpp/string/byte/islower) | Kiểm tra có là ký tự viết thường |
| isupper | Kiểm tra có là ký tự viết hoa |
| isdigit | Kiểm tra có là số |
| [isxdigit](http://en.cppreference.com/w/cpp/string/byte/isxdigit) | Kiểm tra có là ký tự hexa |
| iscntrl | Kiểm tra có là ký tự điều khiẻn |
| isgraph | Kiểm tra có là ký tự đồ họa |
| [isspace](http://en.cppreference.com/w/cpp/string/byte/isspace) | Kiểm tra có là khoảng trắng |
| [isblank](http://en.cppreference.com/w/cpp/string/byte/isblank) | Kiểm tra có là ký tự rỗng |
| isprint | Kiểm tra có là ký tự in |
| ispunct | Kiểm tra có là ký tự chấm |
| [iswctype](http://en.cppreference.com/w/cpp/string/wide/iswctype) | Phân loại một ký tự rộng tùy theo giá trị LC\_CTYPE |
| [wctype](http://en.cppreference.com/w/cpp/string/wide/wctype) | Tìm mội loại ký tự trong dạng hiện tại |

1. Hàm thao tác với ký tự:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [towlower](http://en.cppreference.com/w/cpp/string/wide/towlower) | Chuyển ký tự rộng thành ký tự viết thường |
| [towupper](http://en.cppreference.com/w/cpp/string/wide/towupper) | Chuyển ký tự rộng thành ký tự viết hoa |
| towctrans | Chuyển đổi ký tự khuôn mẫu tùy theo giá trị LC\_CTYPE |
| wctrans | Tìm dạng ký tự khuôn mẫu tại vị trí truy cập hiện tại |

1. Hàm chuyển đổi sang số:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [wcstol](http://en.cppreference.com/w/cpp/string/wide/wcstol)  [wcstoll](http://en.cppreference.com/w/cpp/string/wide/wcstol) | Chuyển thành số nguyên |
| [wcstoul wcstoull](http://en.cppreference.com/w/cpp/string/wide/wcstoul) | Chuyển thành số nguyên dương |
| wcstof  wcstod wcstold | Chuyển thành số thực dấu chấm động |
| [wcstoimax wcstoumax](http://en.cppreference.com/w/cpp/string/wide/wcstoimax) | Chuyển thanh std::intmax\_t hoặc std::uintmax\_t |

1. Hàm thao tác với chuỗi:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [wcscpy](http://en.cppreference.com/w/cpp/string/wide/wcscpy) | Copy sang vào đối tượng khác cùng loại |
| [wcsncpy](http://en.cppreference.com/w/cpp/string/wide/wcsncpy) | Copy một số lượng ký tự rộng xác định |
| [wcscat](http://en.cppreference.com/w/cpp/string/wide/wcscat) | Nối chuỗi |
| [wcsncat](http://en.cppreference.com/w/cpp/string/wide/wcsncat) | Nối chuỗi với n ký tự xác định |
| [wcsxfrm](http://en.cppreference.com/w/cpp/string/wide/wcsxfrm) | So sánh 2 chuỗi |

1. Hàm truy vấn:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [wcslen](http://en.cppreference.com/w/cpp/string/wide/wcslen) | Trả ra độ dài chuỗi |
| [wcscmp](http://en.cppreference.com/w/cpp/string/wide/wcscmp) | So sánh 2 chuỗi |
| [wcsncmp](http://en.cppreference.com/w/cpp/string/wide/wcsncmp) | So sánh 2 chuỗi với số ký tự nhất định |
| [wcscoll](http://en.cppreference.com/w/cpp/string/wide/wcscoll) | So sánh 2 chuỗi phù hợp với vị trí truy cập hiện tại |
| [wcschr](http://en.cppreference.com/w/cpp/string/wide/wcschr) | Tìm vị trí xuất hiện đầu tiên của chuỗi |
| [wcsrchr](http://en.cppreference.com/w/cpp/string/wide/wcsrchr) | Tìm vị trí xuất hiện cuối cùng của chuỗi |
| [wcsspn](http://en.cppreference.com/w/cpp/string/wide/wcsspn) | Trả về độ dài tối đa của đoạn ban đầu chỉ có các ký tự được tìm thấy trong một chuỗi rộng |
| [wcscspn](http://en.cppreference.com/w/cpp/string/wide/wcscspn) | Trả về độ dài tối đa của đoạn ban đầu chỉ có các ký tự không được tìm thấy trong một chuỗi rộng |
| [wcspbrk](http://en.cppreference.com/w/cpp/string/wide/wcspbrk) | Tìm vị trí đầu tiền của ký tự rộng trong chuỗi rộng |
| [wcsstr](http://en.cppreference.com/w/cpp/string/wide/wcsstr) | Tìm vị trí đầu tiên của 1 chuỗi rộng trong 1 chuỗi rộng |
| [wcstok](http://en.cppreference.com/w/cpp/string/wide/wcstok) | Tìm vị trí dấu hiệu tiếp theo trong chuỗi rộng |

1. Hàm thao tác với mảng ký tự wide:

|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| [wmemcpy](http://en.cppreference.com/w/cpp/string/wide/wmemcpy) | Sao chép một số lượng ký tự nhất định của 2 mảng không chồng chéo |
| [wmemmove](http://en.cppreference.com/w/cpp/string/wide/wmemmove) | Sao chép một số lượng ký tự nhất định của 2 mảng chồng chéo |
| [wmemcmp](http://en.cppreference.com/w/cpp/string/wide/wmemcmp) | so sánh một số ký tự nhất định giữa 2 mảng |
| [wmemchr](http://en.cppreference.com/w/cpp/string/wide/wmemchr) | Tìm vị trí xuất hiện đầu tiên của 1 chuỗi trong 1 chuỗi |
| [wmemset](http://en.cppreference.com/w/cpp/string/wide/wmemset) | Sao chép ký tự rộng cho từng ký tự trong mảng |

1. Hàm kiểu mẫu:

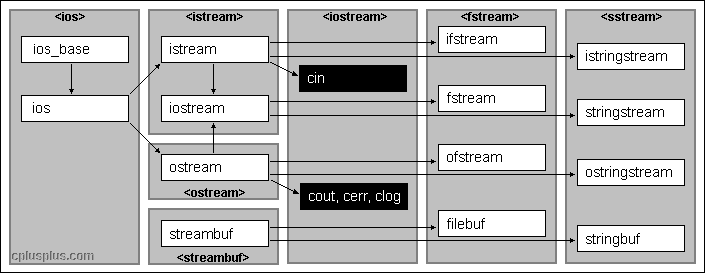
|  |  |
| --- | --- |
| Tên hàm | Chức năng |
| wctrans\_t | Kiểu giữ địa chỉ ký tự khuôn mẫu |
| wctype\_t | Kiểu giữ địa chỉ ký tự phân loại |
| wint\_t | Kiểu số nguyên có thể giữ bất kỳ ký tự rộng hợp lệ và có thể thêm ít nhất một giá trị |

1. Macros:

|  |  |
| --- | --- |
| Tên | Giá trị |
| WEOF | Một dữ liệu phi ký tự có thể dùng để chỉ ra lỗi |
| WCHAR\_MIN | Giá trị hợp lệ nhỏ nhất của wchar\_t |
| WCHAR\_MAX | Giá trị hợp lệ lớn nhất của wchar\_t |

1. **Thư viện nhập xuất (*IOstreams* libraries):**
2. Khái niệm và các thành phần chính:

Input và output streams không được định nghĩa như là một phần của core C++, nhưng nó được cung cấp thông qua C++ standard library (được định nghĩa bên trong namespace std). Nó cung cấp các cơ chế để thực hiện các tác vụ input, output từ đơn giản đến phức tạp. Dưới đây là sơ đồ tổng thể các stream trong C++ được định nghĩa như thế nào:



Ứng với mỗi hình chữ nhật màu trắng trong hình trên là một class trong C++. Các stream trong C++ đều có quan hệ thừa kế với class ios. Và các đối tượng std::cin, std::cout, ... đều được khai báo bên trong file iostream, do đó khi khai báo thư viện iostream thì có thể dùng các đối tượng này.

Đa phần các thao tác liên quan đến Input/Output trong C++ đều được thực thi với các stream. Một cách trừu tượng, một stream là một chuỗi các kí tự được truy cập một cách tuần tự. Theo thời gian, một stream có thể sản xuất hoặc tiêu thụ một lượng dữ liệu không giới hạn.

Chia ra làm 2 loại stream chính: Input và Output.

* Input stream dùng để lưu giữ dữ liệu đầu vào từ data producer, như là bàn phím, file, hoặc một network. Ví dụ, một người dùng ấn một phím khi chương trình hiện tại không yêu cầu bất kỳ dữ liệu đầu vào nào, nó sẽ được lưu tạm thời trong Input stream.
* Output stream dùng để lưu giữ dữ liệu đầu ra cho một data consumer nào đó, như là màn hình, file hoặc printer. Khi chúng ta muốn ghi dữ liệu lên một output device, có thể lúc này device chưa chấp nhận dữ liệu ngay, dữ liệu sẽ được lưu tạm thời trong Output stream.

File vừa có thể là một Input source, vừa có thể là một Output source.

Các class istream và ostream được định nghĩa liên kết mặc định với các standard Input/Output.

* Class istream được sử dụng để trao đổi dữ liệu với các Standard Input streams. Với các input stream, extraction operator (>>) được dùng để xóa bỏ các giá trị trong stream. Điều này có nghĩa là khi người dùng ấn một phím trên bàn phím, key code sẽ được lưu giữ trong input stream, chương trình khác sẽ trích xuất dữ liệu trong input stream ra để sử dụng.
* Class ostream được sử dụng để trao đổi dữ liệu với các Standard Output streams. Với các output stream, insertion operator (<<) được dùng để thêm các giá trị vào trong stream. Điều này có nghĩa là khi các bạn đưa dữ liệu vào trong output stream, các thiết bị đầu ra sẽ sử dụng chúng.

1. Input/Output với file trong C++:

Như đã nhắc đến ở trên, File là một đối tượng có thể được dùng làm Input source, cũng có thể dùng làm Output source. Ngôn ngữ C++ đã định nghĩa sẵn cho chúng ta các class stream dùng để thao tác Input và Output trên File như ifstream (Input file stream), ofstream (Output file stream). Những đối tượng tạo ra từ các class này sẽ đóng vai trò như một stream trung gian để truyền dữ liệu tuần tự vào file hoặc từ file ra.

* Extraction operator (>>) dùng cho các đối tượng của class ifstream dùng để trích xuất dữ liệu trong file đưa vào chương trình.
* Insertion operator (<<) dùng cho các đối tượng của class ofstream dùng để đưa dữ liệu trong chương trình vào lưu trữ trong file.

Do class ifstream được thừa kế từ class istream nên cách hoạt động của các đối tượng của class ifstream cũng giống như đối tượng std::cin, và các đối tượng của class ofstream cũng hoạt động tương tự như đối tượng std::cout của class ostream, nên mình nghĩ trong các bài học tiếp theo các bạn có thể nắm bắt được khá nhanh các thao tác trên file sử dụng các stream này.

1. Stream cho chuỗi:

Các đối tượng của các class stream đóng vai trò bộ truyền dẫn dữ liệu, là đối tượng trung gian giữa chương trình và các thiết bị (hoặc file). Bên cạnh đó, ngôn ngữ C++ còn định nghĩa một số class stream khác giúp chúng ta làm việc với std::string. Do std::string cũng là một kiểu dữ liệu tuần tự, nên sử dụng một stream để truyền dữ liệu vào cho std::string sẽ thuận tiện hơn.

Stream cho std::string mà ngôn ngữ C++ cung cấp là class stringstream được định nghĩa trong thư viện sstream.

1. Các lớp thành phần:

|  |  |
| --- | --- |
| Tên lớp | Đặc điểm |
| ios\_base | Lớp cơ bản của stream |
| ios | Lớp cơ bản của stream |
| istream | Stream nhập |
| ostream | Stream xuất |
| iostream | Stream nhập xuất |
| ifstream | File stream nhập |
| ofstream | File stream xuất |
| fstream | File stream nhập xuất |
| istringstream | String stream nhập |
| ostringstream | String stream xuất |
| stringstream | String stream nhập xuất |
| streambuf | Lớp bộ đệm cơ bản của stream |
| filebuf | Bộ đệm file stream |
| stringbuf | Bộ đệm string stream |

1. Các đối tượng:

|  |  |
| --- | --- |
| Tên | Chức năng |
| [cin](http://en.cppreference.com/w/cpp/io/cin" \o "cpp/io/cin)  [wcin](http://en.cppreference.com/w/cpp/io/cin" \o "cpp/io/cin) | đọc từ các tiêu chuẩn C đầu vào dòng stdin |
| [cout](http://en.cppreference.com/w/cpp/io/cout" \o "cpp/io/cout)  [wcout](http://en.cppreference.com/w/cpp/io/cout" \o "cpp/io/cout) | ghi từ các tiêu chuẩn C đầu vào dòng stdout |
| [cerr](http://en.cppreference.com/w/cpp/io/cerr" \o "cpp/io/cerr)  [wcerr](http://en.cppreference.com/w/cpp/io/cerr" \o "cpp/io/cerr) | viết với tiêu chuẩn C lỗi stream stderr, không có bộ đệm |
| [clog](http://en.cppreference.com/w/cpp/io/clog" \o "cpp/io/clog)  [wclog](http://en.cppreference.com/w/cpp/io/clog" \o "cpp/io/clog) | viết với tiêu chuẩn C lỗi stream stderr. |

1. Loại:

|  |  |
| --- | --- |
| Tên | Đặc điểm |
| fpos | Vị trí lớp stream |
| streamoff | Loại stream bổ sung |
| streampos | vị trí stream |
| streamsize | kích thước stream |

1. Các thao tác:

|  |  |
| --- | --- |
| Tên | Chức năng |
| boolalpha | Giá trị bool chữ số |
| dec | Sử dụng hệ cơ số 10 |
| endl | Chèn dòng mới và flush |
| ends | Chèn ký tự null |
| fixed | Sửa ký hiệu số thực dấu chấm động |
| flush | Flush bộ đệm stream |
| hex | Sử dụng hệ cơ số 16 |
| internal | Điều chỉnh đặc điểm bằng cách chèn ký tự vào vị trí bên trong |
| left | Điều chỉnh đầu ra bên trái |
| noboolalpha | Không có giá trị bool chữ số |
| noshowbase | Không hiển thị tiền tố của hệ cơ số |
| noshowpoint | Không hiển thị dấu thập phân |
| noshowpos | Không hiển thị dấu dương |
| noskipws | Không bỏ qua khoảng trắng |
| nounitbuf | Không tự động flush sau khi chèn |
| nouppercase | Không tạo ra ký tự hoa |
| oct | Sử dụng hệ cơ số 8 |
| resetiosflags | Reset cờ định dạng |
| right | Điều chỉnh đầu ra bên phải |
| scientific | Sử dụng ký hiệu dấu chấm dộng khoa học |
| setbase | Thiết lập cờ đặc tính |
| setfill | Thiết lập ký tự lấp đẩy |
| Setio000sflags | Thiếp lập cờ định dạng |
| setprecision | Thiết lập độ chính xác của số thập phân |
| setw | Thiết lập độ rộng dãy số |
| showbase | Hiển thị tiền tố cơ bản của số |
| showpoint | Hiển thị dấu thập phân |
| showpos | Hiển thị dấu dương |
| skipws | Bỏ qua khoảng tráng |
| unitbuf | Flush bộ đệm sau khi chèn |
| uppercase | Tạo ra ký tự hoa |
| ws | Giải nén khoảng trắng |

1. **Các thuật toán (*Algorithms* STL):**
2. Khái niệm:

STL Algorithm cung cấp một số thuật toán cơ bản để thao tác với các container class. Những thuật toán thường được sử dụng như search, sort, insert, reoder, remove, copy... tất cả đều được sử dụng để thao tác trên các container.

Lưu ý: Các thuật toán này được cài đặt như những hàm có phạm vi global sử dụng các Iterator. Điều này có nghĩa các thuật toán này chỉ cần cài đặt một lần và nó được sử dụng cho các container chứa một tập hợp các iterator.

Khi chúng ta sử dụng các thuật toán này, có thể sẽ giúp chương trình hoạt động nhanh hơn, nhưng cũng có thể có trường hợp làm cho chương trình không hoạt động, lặp vô hạn hoặc có hiệu suất thấp.

Để sử dụng STL algorithm, chỉ cần include thư viện algorithm vào file chương trình là được.

1. Các thao tác không làm thay đổi dữ liệu:

|  |  |
| --- | --- |
| Tên | Chức năng |
| [all\_of](http://en.cppreference.com/w/cpp/algorithm/all_any_none_of" \o "cpp/algorithm/all any none of)  [any\_of](http://en.cppreference.com/w/cpp/algorithm/all_any_none_of" \o "cpp/algorithm/all any none of)  [none\_of](http://en.cppreference.com/w/cpp/algorithm/all_any_none_of" \o "cpp/algorithm/all any none of) | Kiểm tra nếu một tính chất là đúng cho tất cả, bất kỳ hoặc không |
| [for\_each](http://en.cppreference.com/w/cpp/algorithm/for_each) | Thiết đặc một thao tác cho mọi phần tử |
| [for\_each\_n](http://en.cppreference.com/w/cpp/algorithm/for_each_n) | Thiết đặc một thao tác cho n phần tử |
| [count](http://en.cppreference.com/w/cpp/algorithm/count" \o "cpp/algorithm/count)  [count\_if](http://en.cppreference.com/w/cpp/algorithm/count" \o "cpp/algorithm/count) | Đếm số phần tử thoải một số tính chất nhất định |
| mismatch | Tìm vị trí đầu tiên mà 2 dãy khác nhau |
| [equal](http://en.cppreference.com/w/cpp/algorithm/equal) | Chỉ ra 2 dãy giống nhau |
| [find](http://en.cppreference.com/w/cpp/algorithm/find" \o "cpp/algorithm/find)  [find\_if find\_if\_not](http://en.cppreference.com/w/cpp/algorithm/find" \o "cpp/algorithm/find) | Tìm phần tử đầu tiên thỏa tính chất |
| find\_end | Tìm phần tử cuối cùng thỏa tính chất |
| find\_first\_of | Tìm kiếm bất kỳ phần tử nào trong tập hợp |
| [adjacent\_find](http://en.cppreference.com/w/cpp/algorithm/adjacent_find) | Tìm vị trí đầu tiên có 2 phần tử liền kề bằng nhau(hoặc thỏa tính chất nào đó) |
| [search](http://en.cppreference.com/w/cpp/algorithm/search) | Tìm kiếm trên một dãy các phần tử |
| search\_n | Đến số lần mà ít nhất 2 số liền kề giống nhau |

1. Các thao tác làm thay đổi dữ liệu:

|  |  |
| --- | --- |
| Tên | Chức năng |
| [copy](http://en.cppreference.com/w/cpp/algorithm/copy" \o "cpp/algorithm/copy)  [copy\_if](http://en.cppreference.com/w/cpp/algorithm/copy" \o "cpp/algorithm/copy) | Sao chép phần tử |
| [copy\_n](http://en.cppreference.com/w/cpp/algorithm/copy_n) | Sao chép phần tử theo số lượng phần tử |
| [copy\_backward](http://en.cppreference.com/w/cpp/algorithm/copy_backward) | Sao chép phần tử theo từ vị trí đã chọn về sau |
| [move](http://en.cppreference.com/w/cpp/algorithm/move) | Dịch chuyển phần tử |
| [move\_backward](http://en.cppreference.com/w/cpp/algorithm/move_backward) | Dịch chuyển phần tử từ vị trí đã chọn về sau |
| [fill](http://en.cppreference.com/w/cpp/algorithm/fill) | Gán các phần tử bằng giá trị đã cho |
| fill\_n | Gán một số phần tử bằng giá trị đã cho |
| [transform](http://en.cppreference.com/w/cpp/algorithm/transform) | Áp dụng chức năng cho các phần tử |
| [generate](http://en.cppreference.com/w/cpp/algorithm/generate) | Lưu kết quả của 1 hàm trong một phạm vi |
| [generate\_n](http://en.cppreference.com/w/cpp/algorithm/generate_n) | Lưu kết quả của n-chức năng trong một phạm vi |
| [remove remove\_if](http://en.cppreference.com/w/cpp/algorithm/remove) | loại bỏ các phần tử đáp ứng các tiêu chí cụ thể |
| remove\_copy remove\_copy\_if | loại bỏ và sao chép các phần tử đáp ứng các tiêu chí cụ thể |
| [replace replace\_if](http://en.cppreference.com/w/cpp/algorithm/replace) | Sao chép các phần tử đáp ứng các tiêu chí cụ thể |
| [replace\_copy replace\_copy\_if](http://en.cppreference.com/w/cpp/algorithm/replace_copy) | Thay thế các phần tử đáp ứng các tiêu chí cụ thể |
| swap | Chuyển đổi giá trị của 2 đối tượng |
| swap\_ranges | Chuyển đổi giá trị của 2 chuỗi đối tượng |
| [iter\_swap](http://en.cppreference.com/w/cpp/algorithm/iter_swap) | Hoán đổi các yếu tố được trỏ đến bởi hai vòng lặp |
| [reverse](http://en.cppreference.com/w/cpp/algorithm/reverse) | Đảo ngược thứ tự của các yếu tố trong một phạm vi |
| [reverse\_copy](http://en.cppreference.com/w/cpp/algorithm/reverse_copy) | Đảo ngược và sao chép thứ tự của các yếu tố trong một phạm vi |
| [rotate](http://en.cppreference.com/w/cpp/algorithm/rotate) | Quay thứ tự của các yếu tố trong một phạm vi |
| rotate\_copy | Sao chép và quay thứ tự của các yếu tố trong một phạm vi |
| [random\_shuffle shuffle](http://en.cppreference.com/w/cpp/algorithm/random_shuffle) | Sắp xếp ngẫu nhiền lại các phần tử |
| [sample](http://en.cppreference.com/w/cpp/algorithm/sample) | Chọn ngẫu nhiên n phần tử |
| [unique](http://en.cppreference.com/w/cpp/algorithm/unique) | Xóa các phần tử bị trung |
| [unique\_copy](http://en.cppreference.com/w/cpp/algorithm/unique_copy) | Tạo ra một bản sao của một số phạm vi của các thành phần không bị trùng |

1. Thao tác phân vùng:

|  |  |
| --- | --- |
| Tên | Chức năng |
| [is\_partitioned](http://en.cppreference.com/w/cpp/algorithm/is_partitioned) | Xác định nếu phạm vi được phân chia bởi các vị ngữ cho |
| [partition](http://en.cppreference.com/w/cpp/algorithm/partition) | Chia các phần tử thành 2 nhóm |
| [partition\_copy](http://en.cppreference.com/w/cpp/algorithm/partition_copy) | Sao chép các thành phần phân chia thành 2 nhóm |
| [stable\_partition](http://en.cppreference.com/w/cpp/algorithm/stable_partition) | Chia các yếu tố thành hai nhóm trong khi vẫn giữ trật tự tương đối của chúng |
| [partition\_point](http://en.cppreference.com/w/cpp/algorithm/partition_point) | Đặt các điểm phân vùng của một phạm vi phân vùng |

1. Thao tác sắp xếp:

|  |  |
| --- | --- |
| Tên | Chức năng |
| [is\_sorted](http://en.cppreference.com/w/cpp/algorithm/is_sorted) | Kiểm tra đã sắp xếp tăng dần hay chưa |
| [is\_sorted\_until](http://en.cppreference.com/w/cpp/algorithm/is_sorted_until) | Tình dãy phần tử được sắp xếp dài nhất |
| [sort](http://en.cppreference.com/w/cpp/algorithm/sort) | Sắp xếp tăng dần |
| partial\_sort | Sắp xếp n phần tử đầu tiên |
| [partial\_sort\_copy](http://en.cppreference.com/w/cpp/algorithm/partial_sort_copy) | Sao chép và sắp xếp lại các phần tử |
| [stable\_sort](http://en.cppreference.com/w/cpp/algorithm/stable_sort) | Sắp xếp một loạt các yếu tố trong khi vẫn giữ trật tự giữa các yếu tố bình đẳng |
| [nth\_element](http://en.cppreference.com/w/cpp/algorithm/nth_element) | một phần sắp xếp các định phạm vi nhất định chắc chắn rằng nó được phân chia bởi các nguyên tố |

1. Thao toán tìm kiếm nhị phân:

|  |  |
| --- | --- |
| Tên | Chức năng |
| [lower\_bound](http://en.cppreference.com/w/cpp/algorithm/lower_bound) | Trả ra một iterator đến phần tử không nhỏ hơn giá trị của nó |
| [upper\_bound](http://en.cppreference.com/w/cpp/algorithm/upper_bound) | Trả ra một iterator đến phần tử lớn hơn giá trị của nó |
| [binary\_search](http://en.cppreference.com/w/cpp/algorithm/binary_search) | Xác định nếu một phần tử tồn tại trong một phạm vi nhất định |
| [equal\_range](http://en.cppreference.com/w/cpp/algorithm/equal_range) | Trả ra khoảng các phần tử thoải mãn một yếu tố cụ thể |

1. Thao tác thiết đặt:

|  |  |
| --- | --- |
| Tên | Chức năng |
| [merge](http://en.cppreference.com/w/cpp/algorithm/merge) | Trộn 2 mảng đã sắp xếp |
| [inplace\_merge](http://en.cppreference.com/w/cpp/algorithm/inplace_merge) | Trộn hai mảng tại chỗ |
| [includes](http://en.cppreference.com/w/cpp/algorithm/includes) | Trả về true nếu một tập hợp là một tập hợp con của tập hợp khác |
| set\_difference | Tính toán sự khác nhau giữa 2 tập hợp |
| [set\_intersection](http://en.cppreference.com/w/cpp/algorithm/set_intersection) | Tính toán các giao điểm của 2 tập hợp |
| set\_symmetric\_difference | Tính toán sự khác biệt đối xứng giữa hai bộ |
| set\_union | Tính toán các thành phần của 2 bộ |

1. Thao tác vun đống:

|  |  |
| --- | --- |
| Tên | Chức năng |
| [is\_heap](http://en.cppreference.com/w/cpp/algorithm/is_heap) | Kiểm tra nếu chuỗi đã cho là đống lớn nhấp |
| [is\_heap\_until](http://en.cppreference.com/w/cpp/algorithm/is_heap_until) | Tìm chuỗi các phần tử tạo thành một đống lớp nhất |
| [make\_heap](http://en.cppreference.com/w/cpp/algorithm/make_heap) | tạo ra một đống tối đa hiện có của một loạt các phần tử |
| push\_heap | Thêm phần tử vào đống dài nhất |
| [pop\_heap](http://en.cppreference.com/w/cpp/algorithm/pop_heap) | Xóa phần tử lớn nhất ra khỏi đống lớn nhất |
| [sort\_heap](http://en.cppreference.com/w/cpp/algorithm/sort_heap) | Biến một đống tối đa thành một loạt các phần tử được sắp xếp theo thứ tự tăng dần |

1. Thao tác min/max:

|  |  |
| --- | --- |
| Tên | Chức năng |
| [max](http://en.cppreference.com/w/cpp/algorithm/max) | Trả về lớn hơn các giá trị đã cho |
| [max\_element](http://en.cppreference.com/w/cpp/algorithm/max_element) | Trả về phần tử lớn nhất |
| [min](http://en.cppreference.com/w/cpp/algorithm/min) | Trả về nhỏ hơn các giá trị đã cho |
| [min\_element](http://en.cppreference.com/w/cpp/algorithm/min_element) | Trả về phần tử nhỏ nhất |
| [minmax](http://en.cppreference.com/w/cpp/algorithm/minmax) | Trả về giá trị lớn hơn và bé hơn của 2 giá trị |
| [minmax\_element](http://en.cppreference.com/w/cpp/algorithm/minmax_element) | Trả về giá trị lớn nhất và bé nhất của 2 giá trị trong 1 khoảng |
| [clamp](http://en.cppreference.com/w/cpp/algorithm/clamp) | Kẹp một giá trị giữa một cặp giá trị ranh giới |
| [lexicographical\_compare](http://en.cppreference.com/w/cpp/algorithm/lexicographical_compare) | Trả về true nếu một phạm vi là thứ tự từ điển nhỏ hơn khác |
| [is\_permutation](http://en.cppreference.com/w/cpp/algorithm/is_permutation) | Xác định nếu một trình tự là một hoán vị của dãy khác |
| [next\_permutation](http://en.cppreference.com/w/cpp/algorithm/next_permutation) | Tạo ra sự hoán vị tự từ điển lớn hơn tiếp theo của một loạt các giá trị |
| prev\_permutation | tạo ra sự hoán vị tự từ điển nhỏ hơn tiếp theo của một loạt các giá trị |

1. Thao tác số học:

|  |  |
| --- | --- |
| Tên | Chức năng |
| [iota](http://en.cppreference.com/w/cpp/algorithm/iota) | Chuyển từ dạng int sang dạng ký tự |
| [accumulate](http://en.cppreference.com/w/cpp/algorithm/accumulate) | Tính tổng của 1 chuỗi các phần tử |
| [inner\_product](http://en.cppreference.com/w/cpp/algorithm/inner_product) | Tính giá trị bên trong của 2 dãy nguyên tố |
| [adjacent\_difference](http://en.cppreference.com/w/cpp/algorithm/adjacent_difference) | Tính toán sự khác biệt giữa các yếu tố cạnh nhau trong một phạm vi |
| partial\_sum | Tính tổng một phần của một loạt các yếu tố |
| [reduce](http://en.cppreference.com/w/cpp/algorithm/reduce) | Tương tự [accumulate](http://en.cppreference.com/w/cpp/algorithm/accumulate) |
| [exclusive\_scan](http://en.cppreference.com/w/cpp/algorithm/exclusive_scan) | Tương tự partial\_sum |
| [inclusive\_scan](http://en.cppreference.com/w/cpp/algorithm/inclusive_scan) | Tương tự partial\_sum |
| [transform\_reduce](http://en.cppreference.com/w/cpp/algorithm/transform_reduce) | Áp dụng một hàm tử, sau đó giảm trong trật tự |
| [transform\_exclusive\_scan](http://en.cppreference.com/w/cpp/algorithm/transform_exclusive_scan) | Áp dụng một hàm tử, sau đó tính toán quét 1 phần |
| [transform\_inclusive\_scan](http://en.cppreference.com/w/cpp/algorithm/transform_inclusive_scan) | Áp dụng một hàm tử, sau đó tính toán quét toàn diện |

1. Thao tác trên bộ nhớ chưa được định hình:

|  |  |
| --- | --- |
| Tên | Chức năng |
| uninitialized\_copy | Sao chép một dãy các phần tử vào một vùng nhớ không xác định |
| [uninitialized\_copy\_n](http://en.cppreference.com/w/cpp/memory/uninitialized_copy_n) | Sao chép các phần tử vào một vùng nhớ không xác định |
| [uninitialized\_fill](http://en.cppreference.com/w/cpp/memory/uninitialized_fill) | Bản sao một đối tượng đến một khu vực chưa được khởi tạo bộ nhớ, được xác định bởi một khoảng |
| [uninitialized\_fill\_n](http://en.cppreference.com/w/cpp/memory/uninitialized_fill_n) | Bản sao một đối tượng đến một khu vực chưa được khởi tạo bộ nhớ, được xác định bởi vị trí bắt đầu và số lượng |
| [uninitialized\_move](http://en.cppreference.com/w/cpp/memory/uninitialized_move) | Di chuyển một loạt các đối tượng đến một khu vực chưa được khởi tạo bộ nhớ |
| [uninitialized\_move\_n](http://en.cppreference.com/w/cpp/memory/uninitialized_move_n) | Di chuyển một số các đối tượng đến một khu vực chưa được khởi tạo bộ nhớ |

1. Thư viện C:

|  |  |
| --- | --- |
| Tên | Chức năng |
| qsort | Sắp xếp một loạt các yếu tố với loại không xác định |
| [bsearch](http://en.cppreference.com/w/cpp/algorithm/bsearch) | Tìm kiếm một mảng cho một phần tử kiểu không xác định |

1. **Thông tin bổ sung:**

* **std::char\_traits** : Lớp char\_traits là một lớp mẫu mà tóm tắt hoạt động của đối tượng và chuỗi cơ bản cho một loại đối tượng nhất định. Các thiết lập hoạt động được xác định là như vậy mà các thuật toán hầu như luôn luôn có thể được thực hiện trong các ràng buộc của thuật toán. Vì thế, có thể sử dụng các thuật toán như vậy với hầu như bất kỳ đối tượng có thể hay kiểu chuỗi, chỉ bằng cách cung cấp tùy chỉnh lớp char\_traits. Tìm hiểu thêm tại: <http://en.cppreference.com/w/cpp/algorithm>.
* **iterators** : iterator là đối tượng nào đó, chỉ tới một phần tử trong tập hợp (như một mảng hoặc một container), có khả năng duyệt qua các phần tử trong tập hợp đó sử dụng một tập hợp các toán tử (với ít nhất là tăng (+ +) và ttham chiếu). Tìm hiểu thêm tại: <http://www.cplusplus.com/reference/iterator/>
* **literals** : Trong khoa học máy tính, literals là một ký hiệu đại diện cho một giá trị cố định trong mã nguồn. Hầu như tất cả các ngôn ngữ lập trình có ký hiệu cho các giá trị nguyên tử như số nguyên, số thập phân, và các chuỗi, và thường cho các phép toán luận và ký tự; cũng có một số ký hiệu cho các yếu tố của các loại được liệt kê và giá trị hợp chất như mảng, hồ sơ, và các đối tượng. Một chức năng vô danh là một literals cho các loại chức năng. Tìm hiểu thêm tại:

<https://en.wikipedia.org/wiki/Literal_(computer_programming)>

* **wide character** : Là một kiểu dữ liệu ký tự có kích thước lớn hơn bộ dữ liệu ký tự truyền thống (8 bit) việc tăng ký thước lớn hơn cho phép sử dụng bộ ký tự được mã hóa lớn hơn. Tìm hiểu thêm tại: <https://en.wikipedia.org/wiki/Wide_character>
* **flush** : Là một hàm chức năng thuộc std::ostream dùng để đồng bộ dòng đệm và chuỗi đầu ra. Đối với đối tượng dòng đệm mà thực hiện bộ đệm trung gian, chức năng này yêu cầu tất cả các đối tượng phải được ghi vào trình tự kiểm soát. Hành vi của nó tương đương với gọi hàm thành viên flush. Tìm hiểu thêm tại: <http://www.cplusplus.com/reference/ostream/flush-free/>
* **control character:**
* **Các nguồn tài liệu tham khảo chính:**
* <http://www.cplusplus.com/>.
* <http://en.cppreference.com/>.
* <https://www.stdio.vn>.
* <https://daynhauhoc.com>
* <https://msdn.microsoft.com>
* **Demo:**
* std::basic\_string\_view và null-terminated wide byte string sẽ không có phần demo vì các thao tác và hàm được cung cấp trong std::basic\_string\_view và null-terminated wide byte string lần lượt tương tự các đặc điểm và chức năng trong std::basic\_string và null-terminated byte string.