**Standard Template Library: Algorithms**

The header <algorithm> defines a collection of functions especially designed to be used on ranges of elements.  
  
A range is any sequence of objects that can be accessed through iterators or pointers, such as an array or an instance of some of the [STL containers](http://www.cplusplus.com/stl). Notice though, that algorithms operate through iterators directly on the values, not affecting in any way the structure of any possible container (it never affects the size or storage allocation of the container).

**Functions in <algorithm>**

**Non-modifying sequence operations**:

[**all\_of**](http://www.cplusplus.com/reference/algorithm/all_of/)

Test condition on all elements in range (function template )

[**any\_of**](http://www.cplusplus.com/reference/algorithm/any_of/)

Test if any element in range fulfills condition (function template )

[**none\_of**](http://www.cplusplus.com/reference/algorithm/none_of/)

Test if no elements fulfill condition (function template )

[**for\_each**](http://www.cplusplus.com/reference/algorithm/for_each/)

Apply function to range (function template )

[**find**](http://www.cplusplus.com/reference/algorithm/find/)

Find value in range (function template )

[**find\_if**](http://www.cplusplus.com/reference/algorithm/find_if/)

Find element in range (function template )

[**find\_if\_not**](http://www.cplusplus.com/reference/algorithm/find_if_not/)

Find element in range (negative condition) (function template )

[**find\_end**](http://www.cplusplus.com/reference/algorithm/find_end/)

Find last subsequence in range (function template )

[**find\_first\_of**](http://www.cplusplus.com/reference/algorithm/find_first_of/)

Find element from set in range (function template )

[**adjacent\_find**](http://www.cplusplus.com/reference/algorithm/adjacent_find/)

Find equal adjacent elements in range (function template )

[**count**](http://www.cplusplus.com/reference/algorithm/count/) **–** Input = (first, last, value), first and last specify range (inclusive of first and last), O(N)

Count appearances of value in range (function template )

[**count\_if**](http://www.cplusplus.com/reference/algorithm/count_if/) **–** Input = (first, last, pred), pred = Boolean function used as condition, O(N) but pred is called once for each element

Return number of elements in range satisfying condition (function template )

[**mismatch**](http://www.cplusplus.com/reference/algorithm/mismatch/)

Return first position where two ranges differ (function template )

[**equal**](http://www.cplusplus.com/reference/algorithm/equal/)

Test whether the elements in two ranges are equal (function template )

[**is\_permutation**](http://www.cplusplus.com/reference/algorithm/is_permutation/)

Test whether range is permutation of another (function template )

[**search**](http://www.cplusplus.com/reference/algorithm/search/)

Search range for subsequence (function template )

[**search\_n**](http://www.cplusplus.com/reference/algorithm/search_n/)

Search range for elements (function template )

**Modifying sequence operations**:

[**copy**](http://www.cplusplus.com/reference/algorithm/copy/)

Copy range of elements (function template )

[**copy\_n**](http://www.cplusplus.com/reference/algorithm/copy_n/)

Copy elements (function template )

[**copy\_if**](http://www.cplusplus.com/reference/algorithm/copy_if/)

Copy certain elements of range (function template )

[**copy\_backward**](http://www.cplusplus.com/reference/algorithm/copy_backward/)

Copy range of elements backward (function template )

[**move**](http://www.cplusplus.com/reference/algorithm/move/)

Move range of elements (function template )

[**move\_backward**](http://www.cplusplus.com/reference/algorithm/move_backward/)

Move range of elements backward (function template )

[**swap**](http://www.cplusplus.com/reference/algorithm/swap/)

Exchange values of two objects (function template )

[**swap\_ranges**](http://www.cplusplus.com/reference/algorithm/swap_ranges/)

Exchange values of two ranges (function template )

[**iter\_swap**](http://www.cplusplus.com/reference/algorithm/iter_swap/)

Exchange values of objects pointed to by two iterators (function template )

[**transform**](http://www.cplusplus.com/reference/algorithm/transform/)

Transform range (function template )

[**replace**](http://www.cplusplus.com/reference/algorithm/replace/)

Replace value in range (function template )

[**replace\_if**](http://www.cplusplus.com/reference/algorithm/replace_if/)

Replace values in range (function template )

[**replace\_copy**](http://www.cplusplus.com/reference/algorithm/replace_copy/)

Copy range replacing value (function template )

[**replace\_copy\_if**](http://www.cplusplus.com/reference/algorithm/replace_copy_if/)

Copy range replacing value (function template )

[**fill**](http://www.cplusplus.com/reference/algorithm/fill/)

Fill range with value (function template )

[**fill\_n**](http://www.cplusplus.com/reference/algorithm/fill_n/)

Fill sequence with value (function template )

[**generate**](http://www.cplusplus.com/reference/algorithm/generate/)

Generate values for range with function (function template )

[**generate\_n**](http://www.cplusplus.com/reference/algorithm/generate_n/)

Generate values for sequence with function (function template )

[**remove**](http://www.cplusplus.com/reference/algorithm/remove/)

Remove value from range (function template )

[**remove\_if**](http://www.cplusplus.com/reference/algorithm/remove_if/)

Remove elements from range (function template )

[**remove\_copy**](http://www.cplusplus.com/reference/algorithm/remove_copy/)

Copy range removing value (function template )

[**remove\_copy\_if**](http://www.cplusplus.com/reference/algorithm/remove_copy_if/)

Copy range removing values (function template )

[**unique**](http://www.cplusplus.com/reference/algorithm/unique/)

Remove consecutive duplicates in range (function template )

[**unique\_copy**](http://www.cplusplus.com/reference/algorithm/unique_copy/)

Copy range removing duplicates (function template )

[**reverse**](http://www.cplusplus.com/reference/algorithm/reverse/)

Reverse range (function template )

[**reverse\_copy**](http://www.cplusplus.com/reference/algorithm/reverse_copy/)

Copy range reversed (function template )

[**rotate**](http://www.cplusplus.com/reference/algorithm/rotate/)

Rotate left the elements in range (function template )

[**rotate\_copy**](http://www.cplusplus.com/reference/algorithm/rotate_copy/)

Copy range rotated left (function template )

[**random\_shuffle**](http://www.cplusplus.com/reference/algorithm/random_shuffle/)

Randomly rearrange elements in range (function template )

[**shuffle**](http://www.cplusplus.com/reference/algorithm/shuffle/)

Randomly rearrange elements in range using generator (function template )

**Partitions**:

[**is\_partitioned**](http://www.cplusplus.com/reference/algorithm/is_partitioned/)

Test whether range is partitioned (function template )

[**partition**](http://www.cplusplus.com/reference/algorithm/partition/)

Partition range in two (function template )

[**stable\_partition**](http://www.cplusplus.com/reference/algorithm/stable_partition/)

Partition range in two - stable ordering (function template )

[**partition\_copy**](http://www.cplusplus.com/reference/algorithm/partition_copy/)

Partition range into two (function template )

[**partition\_point**](http://www.cplusplus.com/reference/algorithm/partition_point/)

Get partition point (function template )

**Sorting**: O(NlogN)

[**sort**](http://www.cplusplus.com/reference/algorithm/sort/) **–** Input: (first, last, comparator), O(N\*logN), note that comparator is a Boolean function

Sort elements in range (function template )

[**stable\_sort**](http://www.cplusplus.com/reference/algorithm/stable_sort/)

Sort elements preserving order of equivalents (function template )

[**partial\_sort**](http://www.cplusplus.com/reference/algorithm/partial_sort/)

Partially sort elements in range (function template )

[**partial\_sort\_copy**](http://www.cplusplus.com/reference/algorithm/partial_sort_copy/)

Copy and partially sort range (function template )

[**is\_sorted**](http://www.cplusplus.com/reference/algorithm/is_sorted/)

Check whether range is sorted (function template )

[**is\_sorted\_until**](http://www.cplusplus.com/reference/algorithm/is_sorted_until/)

Find first unsorted element in range (function template )

[**nth\_element**](http://www.cplusplus.com/reference/algorithm/nth_element/)– Input: (first, nth, last, comparator), places nth in its sorted position (effectively partitioning array along n)

Sort element in range (function template )

**Binary search** (operating on partitioned/sorted ranges): Inputs = (first, last, value, comparator), O(logN)

[**lower\_bound**](http://www.cplusplus.com/reference/algorithm/lower_bound/) – Return iterator to first element in range which equals value (if value exists) or first element that is greater than value (if value doesn’t exist)

Return iterator to lower bound (function template )

[**upper\_bound**](http://www.cplusplus.com/reference/algorithm/upper_bound/)- Return iterator to last element in range which equals value (if value exists) or first element that is less than value (if value doesn’t exist)

Return iterator to upper bound (function template )

[**equal\_range**](http://www.cplusplus.com/reference/algorithm/equal_range/)

Get subrange of equal elements (function template )

[**binary\_search**](http://www.cplusplus.com/reference/algorithm/binary_search/) – Boolean Function, returns true if value exists

Test if value exists in sorted sequence (function template )

**Merge** (operating on sorted ranges):

[**merge**](http://www.cplusplus.com/reference/algorithm/merge/)

Merge sorted ranges (function template )

[**inplace\_merge**](http://www.cplusplus.com/reference/algorithm/inplace_merge/)

Merge consecutive sorted ranges (function template )

[**includes**](http://www.cplusplus.com/reference/algorithm/includes/)

Test whether sorted range includes another sorted range (function template )

[**set\_union**](http://www.cplusplus.com/reference/algorithm/set_union/)

Union of two sorted ranges (function template )

[**set\_intersection**](http://www.cplusplus.com/reference/algorithm/set_intersection/)

Intersection of two sorted ranges (function template )

[**set\_difference**](http://www.cplusplus.com/reference/algorithm/set_difference/)

Difference of two sorted ranges (function template )

[**set\_symmetric\_difference**](http://www.cplusplus.com/reference/algorithm/set_symmetric_difference/)

Symmetric difference of two sorted ranges (function template )

**Heap**:

[**push\_heap**](http://www.cplusplus.com/reference/algorithm/push_heap/)

Push element into heap range (function template )

[**pop\_heap**](http://www.cplusplus.com/reference/algorithm/pop_heap/)

Pop element from heap range (function template )

[**make\_heap**](http://www.cplusplus.com/reference/algorithm/make_heap/)

Make heap from range (function template )

[**sort\_heap**](http://www.cplusplus.com/reference/algorithm/sort_heap/)

Sort elements of heap (function template )

[**is\_heap**](http://www.cplusplus.com/reference/algorithm/is_heap/)

Test if range is heap (function template )

[**is\_heap\_until**](http://www.cplusplus.com/reference/algorithm/is_heap_until/)

Find first element not in heap order (function template )

**Min/max**:

[**min**](http://www.cplusplus.com/reference/algorithm/min/)

Return the smallest (function template )

[**max**](http://www.cplusplus.com/reference/algorithm/max/)

Return the largest (function template )

[**minmax**](http://www.cplusplus.com/reference/algorithm/minmax/)

Return smallest and largest elements (function template )

[**min\_element**](http://www.cplusplus.com/reference/algorithm/min_element/)

Return smallest element in range (function template )

[**max\_element**](http://www.cplusplus.com/reference/algorithm/max_element/)

Return largest element in range (function template )

[**minmax\_element**](http://www.cplusplus.com/reference/algorithm/minmax_element/)

Return smallest and largest elements in range (function template )

**Other**:

[**lexicographical\_compare**](http://www.cplusplus.com/reference/algorithm/lexicographical_compare/)

Lexicographical less-than comparison (function template )

[**next\_permutation**](http://www.cplusplus.com/reference/algorithm/next_permutation/)

Transform range to next permutation (function template )

[**prev\_permutation**](http://www.cplusplus.com/reference/algorithm/prev_permutation/)

Transform range to previous permutation (function template )

**Vector**

**Member functions**

[**(constructor)**](http://www.cplusplus.com/reference/vector/vector/vector/)

Construct vector (public member function )

[**(destructor)**](http://www.cplusplus.com/reference/vector/vector/~vector/)

Vector destructor (public member function )

[**operator=**](http://www.cplusplus.com/reference/vector/vector/operator=/)

Assign content (public member function )

**Iterators**:

[**begin**](http://www.cplusplus.com/reference/vector/vector/begin/)

Return iterator to beginning (public member function )

[**end**](http://www.cplusplus.com/reference/vector/vector/end/)

Return iterator to end (public member function )

[**rbegin**](http://www.cplusplus.com/reference/vector/vector/rbegin/)

Return reverse iterator to reverse beginning (public member function )

[**rend**](http://www.cplusplus.com/reference/vector/vector/rend/)

Return reverse iterator to reverse end (public member function )

[**cbegin**](http://www.cplusplus.com/reference/vector/vector/cbegin/)

Return const\_iterator to beginning (public member function )

[**cend**](http://www.cplusplus.com/reference/vector/vector/cend/)

Return const\_iterator to end (public member function )

[**crbegin**](http://www.cplusplus.com/reference/vector/vector/crbegin/)

Return const\_reverse\_iterator to reverse beginning (public member function )

[**crend**](http://www.cplusplus.com/reference/vector/vector/crend/)

Return const\_reverse\_iterator to reverse end (public member function )

**Capacity**:

[**size**](http://www.cplusplus.com/reference/vector/vector/size/)

Return size (public member function )

[**max\_size**](http://www.cplusplus.com/reference/vector/vector/max_size/)

Return maximum size (public member function )

[**resize**](http://www.cplusplus.com/reference/vector/vector/resize/)

Change size (public member function )

[**capacity**](http://www.cplusplus.com/reference/vector/vector/capacity/)

Return size of allocated storage capacity (public member function )

[**empty**](http://www.cplusplus.com/reference/vector/vector/empty/)

Test whether vector is empty (public member function )

[**reserve**](http://www.cplusplus.com/reference/vector/vector/reserve/)

Request a change in capacity (public member function )

[**shrink\_to\_fit**](http://www.cplusplus.com/reference/vector/vector/shrink_to_fit/)

Shrink to fit (public member function )

**Element access**:

[**operator[]**](http://www.cplusplus.com/reference/vector/vector/operator%5b%5d/)

Access element (public member function )

[**at**](http://www.cplusplus.com/reference/vector/vector/at/)

Access element (public member function )

[**front**](http://www.cplusplus.com/reference/vector/vector/front/)

Access first element (public member function )

[**back**](http://www.cplusplus.com/reference/vector/vector/back/)

Access last element (public member function )

[**data**](http://www.cplusplus.com/reference/vector/vector/data/)

Access data (public member function )

**Modifiers**:

[**assign**](http://www.cplusplus.com/reference/vector/vector/assign/)

Assign vector content (public member function )

[**push\_back**](http://www.cplusplus.com/reference/vector/vector/push_back/)

Add element at the end (public member function )

[**pop\_back**](http://www.cplusplus.com/reference/vector/vector/pop_back/)

Delete last element (public member function )

[**insert**](http://www.cplusplus.com/reference/vector/vector/insert/)

Insert elements (public member function )

[**erase**](http://www.cplusplus.com/reference/vector/vector/erase/)

Erase elements (public member function )

[**swap**](http://www.cplusplus.com/reference/vector/vector/swap/)

Swap content (public member function )

[**clear**](http://www.cplusplus.com/reference/vector/vector/clear/)

Clear content (public member function )

[**emplace**](http://www.cplusplus.com/reference/vector/vector/emplace/)

Construct and insert element (public member function )

[**emplace\_back**](http://www.cplusplus.com/reference/vector/vector/emplace_back/)

Construct and insert element at the end (public member function )

**Allocator**:

[**get\_allocator**](http://www.cplusplus.com/reference/vector/vector/get_allocator/)

Get allocator (public member function )

**Non-member function overloads**

[**relational operators**](http://www.cplusplus.com/reference/vector/vector/operators/)

Relational operators for vector (function template )

[**swap**](http://www.cplusplus.com/reference/vector/vector/swap-free/)

Exchange contents of vectors (function template )

**Template specializations**

[**vector<bool>**](http://www.cplusplus.com/reference/vector/vector-bool/)

Vector of bool (class template specialization )

**String**

**Member functions**

[**(constructor)**](http://www.cplusplus.com/reference/string/string/string/)

Construct string object (public member function )

[**(destructor)**](http://www.cplusplus.com/reference/string/string/~string/)

String destructor (public member function )

[**operator=**](http://www.cplusplus.com/reference/string/string/operator=/)

String assignment (public member function )

**Iterators**:

[**begin**](http://www.cplusplus.com/reference/string/string/begin/)

Return iterator to beginning (public member function )

[**end**](http://www.cplusplus.com/reference/string/string/end/)

Return iterator to end (public member function )

[**rbegin**](http://www.cplusplus.com/reference/string/string/rbegin/)

Return reverse iterator to reverse beginning (public member function )

[**rend**](http://www.cplusplus.com/reference/string/string/rend/)

Return reverse iterator to reverse end (public member function )

[**cbegin**](http://www.cplusplus.com/reference/string/string/cbegin/)

Return const\_iterator to beginning (public member function )

[**cend**](http://www.cplusplus.com/reference/string/string/cend/)

Return const\_iterator to end (public member function )

[**crbegin**](http://www.cplusplus.com/reference/string/string/crbegin/)

Return const\_reverse\_iterator to reverse beginning (public member function )

[**crend**](http://www.cplusplus.com/reference/string/string/crend/)

Return const\_reverse\_iterator to reverse end (public member function )

**Capacity**:

[**size**](http://www.cplusplus.com/reference/string/string/size/)

Return length of string (public member function )

[**length**](http://www.cplusplus.com/reference/string/string/length/)

Return length of string (public member function )

[**max\_size**](http://www.cplusplus.com/reference/string/string/max_size/)

Return maximum size of string (public member function )

[**resize**](http://www.cplusplus.com/reference/string/string/resize/)

Resize string (public member function )

[**capacity**](http://www.cplusplus.com/reference/string/string/capacity/)

Return size of allocated storage (public member function )

[**reserve**](http://www.cplusplus.com/reference/string/string/reserve/)

Request a change in capacity (public member function )

[**clear**](http://www.cplusplus.com/reference/string/string/clear/)

Clear string (public member function )

[**empty**](http://www.cplusplus.com/reference/string/string/empty/)

Test if string is empty (public member function )

[**shrink\_to\_fit**](http://www.cplusplus.com/reference/string/string/shrink_to_fit/)

Shrink to fit (public member function )

**Element access**:

[**operator[]**](http://www.cplusplus.com/reference/string/string/operator%5b%5d/)

Get character of string (public member function )

[**at**](http://www.cplusplus.com/reference/string/string/at/)

Get character in string (public member function )

[**back**](http://www.cplusplus.com/reference/string/string/back/)

Access last character (public member function )

[**front**](http://www.cplusplus.com/reference/string/string/front/)

Access first character (public member function )

**Modifiers**:

[**operator+=**](http://www.cplusplus.com/reference/string/string/operator+=/)

Append to string (public member function )

[**append**](http://www.cplusplus.com/reference/string/string/append/)

Append to string (public member function )

[**push\_back**](http://www.cplusplus.com/reference/string/string/push_back/)

Append character to string (public member function )

[**assign**](http://www.cplusplus.com/reference/string/string/assign/)

Assign content to string (public member function )

[**insert**](http://www.cplusplus.com/reference/string/string/insert/)

Insert into string (public member function )

[**erase**](http://www.cplusplus.com/reference/string/string/erase/)

Erase characters from string (public member function )

[**replace**](http://www.cplusplus.com/reference/string/string/replace/)

Replace portion of string (public member function )

[**swap**](http://www.cplusplus.com/reference/string/string/swap/)

Swap string values (public member function )

[**pop\_back**](http://www.cplusplus.com/reference/string/string/pop_back/)

Delete last character (public member function )

**String operations**:

[**c\_str**](http://www.cplusplus.com/reference/string/string/c_str/)

Get C string equivalent (public member function )

[**data**](http://www.cplusplus.com/reference/string/string/data/)

Get string data (public member function )

[**get\_allocator**](http://www.cplusplus.com/reference/string/string/get_allocator/)

Get allocator (public member function )

[**copy**](http://www.cplusplus.com/reference/string/string/copy/)

Copy sequence of characters from string (public member function )

[**find**](http://www.cplusplus.com/reference/string/string/find/)

Find content in string (public member function )

[**rfind**](http://www.cplusplus.com/reference/string/string/rfind/)

Find last occurrence of content in string (public member function )

[**find\_first\_of**](http://www.cplusplus.com/reference/string/string/find_first_of/)

Find character in string (public member function )

[**find\_last\_of**](http://www.cplusplus.com/reference/string/string/find_last_of/)

Find character in string from the end (public member function )

[**find\_first\_not\_of**](http://www.cplusplus.com/reference/string/string/find_first_not_of/)

Find absence of character in string (public member function )

[**find\_last\_not\_of**](http://www.cplusplus.com/reference/string/string/find_last_not_of/)

Find non-matching character in string from the end (public member function )

[**substr**](http://www.cplusplus.com/reference/string/string/substr/)

Generate substring (public member function )

[**compare**](http://www.cplusplus.com/reference/string/string/compare/)

Compare strings (public member function )

**Member constants**

[**npos**](http://www.cplusplus.com/reference/string/string/npos/)

Maximum value for size\_t (public static member constant )

**Non-member function overloads**

[**operator+**](http://www.cplusplus.com/reference/string/string/operator+/)

Concatenate strings (function )

[**relational operators**](http://www.cplusplus.com/reference/string/string/operators/)

Relational operators for string (function )

[**swap**](http://www.cplusplus.com/reference/string/string/swap-free/)

Exchanges the values of two strings (function )

[**operator>>**](http://www.cplusplus.com/reference/string/string/operator%3E%3E/)

Extract string from stream (function )

[**operator<<**](http://www.cplusplus.com/reference/string/string/operator%3C%3C/)

Insert string into stream (function )

[**getline**](http://www.cplusplus.com/reference/string/string/getline/)

Get line from stream into string (function )

**List**

**Member functions**

[**(constructor)**](http://www.cplusplus.com/reference/list/list/list/)

Construct list (public member function )

[**(destructor)**](http://www.cplusplus.com/reference/list/list/~list/)

List destructor (public member function )

[**operator=**](http://www.cplusplus.com/reference/list/list/operator=/)

Assign content (public member function )

**Iterators**:

[**begin**](http://www.cplusplus.com/reference/list/list/begin/)

Return iterator to beginning (public member function )

[**end**](http://www.cplusplus.com/reference/list/list/end/)

Return iterator to end (public member function )

[**rbegin**](http://www.cplusplus.com/reference/list/list/rbegin/)

Return reverse iterator to reverse beginning (public member function )

[**rend**](http://www.cplusplus.com/reference/list/list/rend/)

Return reverse iterator to reverse end (public member function )

[**cbegin**](http://www.cplusplus.com/reference/list/list/cbegin/)

Return const\_iterator to beginning (public member function )

[**cend**](http://www.cplusplus.com/reference/list/list/cend/)

Return const\_iterator to end (public member function )

[**crbegin**](http://www.cplusplus.com/reference/list/list/crbegin/)

Return const\_reverse\_iterator to reverse beginning (public member function )

[**crend**](http://www.cplusplus.com/reference/list/list/crend/)

Return const\_reverse\_iterator to reverse end (public member function )

**Capacity**:

[**empty**](http://www.cplusplus.com/reference/list/list/empty/)

Test whether container is empty (public member function )

[**size**](http://www.cplusplus.com/reference/list/list/size/)

Return size (public member function )

[**max\_size**](http://www.cplusplus.com/reference/list/list/max_size/)

Return maximum size (public member function )

**Element access**:

[**front**](http://www.cplusplus.com/reference/list/list/front/)

Access first element (public member function )

[**back**](http://www.cplusplus.com/reference/list/list/back/)

Access last element (public member function )

**Modifiers**:

[**assign**](http://www.cplusplus.com/reference/list/list/assign/)

Assign new content to container (public member function )

[**emplace\_front**](http://www.cplusplus.com/reference/list/list/emplace_front/)

Construct and insert element at beginning (public member function )

[**push\_front**](http://www.cplusplus.com/reference/list/list/push_front/)

Insert element at beginning (public member function )

[**pop\_front**](http://www.cplusplus.com/reference/list/list/pop_front/)

Delete first element (public member function )

[**emplace\_back**](http://www.cplusplus.com/reference/list/list/emplace_back/)

Construct and insert element at the end (public member function )

[**push\_back**](http://www.cplusplus.com/reference/list/list/push_back/)

Add element at the end (public member function )

[**pop\_back**](http://www.cplusplus.com/reference/list/list/pop_back/)

Delete last element (public member function )

[**emplace**](http://www.cplusplus.com/reference/list/list/emplace/)

Construct and insert element (public member function )

[**insert**](http://www.cplusplus.com/reference/list/list/insert/)

Insert elements (public member function )

[**erase**](http://www.cplusplus.com/reference/list/list/erase/)

Erase elements (public member function )

[**swap**](http://www.cplusplus.com/reference/list/list/swap/)

Swap content (public member function )

[**resize**](http://www.cplusplus.com/reference/list/list/resize/)

Change size (public member function )

[**clear**](http://www.cplusplus.com/reference/list/list/clear/)

Clear content (public member function )

**Operations**:

[**splice**](http://www.cplusplus.com/reference/list/list/splice/)

Transfer elements from list to list (public member function )

[**remove**](http://www.cplusplus.com/reference/list/list/remove/)

Remove elements with specific value (public member function )

[**remove\_if**](http://www.cplusplus.com/reference/list/list/remove_if/)

Remove elements fulfilling condition (public member function template )

[**unique**](http://www.cplusplus.com/reference/list/list/unique/)

Remove duplicate values (public member function )

[**merge**](http://www.cplusplus.com/reference/list/list/merge/)

Merge sorted lists (public member function )

[**sort**](http://www.cplusplus.com/reference/list/list/sort/)

Sort elements in container (public member function )

[**reverse**](http://www.cplusplus.com/reference/list/list/reverse/)

Reverse the order of elements (public member function )

**Observers**:

[**get\_allocator**](http://www.cplusplus.com/reference/list/list/get_allocator/)

Get allocator (public member function )

**Non-member function overloads**

[**relational operators (list)**](http://www.cplusplus.com/reference/list/list/operators/)

Relational operators for list (function )

[**swap (list)**](http://www.cplusplus.com/reference/list/list/swap-free/)

Exchanges the contents of two lists (function template )

**Stack**

**Member functions**

[**(constructor)**](http://www.cplusplus.com/reference/stack/stack/stack/)

Construct stack (public member function )

[**empty**](http://www.cplusplus.com/reference/stack/stack/empty/)

Test whether container is empty (public member function )

[**size**](http://www.cplusplus.com/reference/stack/stack/size/)

Return size (public member function )

[**top**](http://www.cplusplus.com/reference/stack/stack/top/)

Access next element (public member function )

[**push**](http://www.cplusplus.com/reference/stack/stack/push/)

Insert element (public member function )

[**emplace**](http://www.cplusplus.com/reference/stack/stack/emplace/)

Construct and insert element (public member function )

[**pop**](http://www.cplusplus.com/reference/stack/stack/pop/)

Remove top element (public member function )

[**swap**](http://www.cplusplus.com/reference/stack/stack/swap/)

Swap contents (public member function )

**Non-member function overloads**

[**relational operators**](http://www.cplusplus.com/reference/stack/stack/operators/)

Relational operators for stack (function )

[**swap (stack)**](http://www.cplusplus.com/reference/stack/stack/swap-free/)

Exchange contents of stacks (public member function )

**Non-member class specializations**

[**uses\_allocator<stack>**](http://www.cplusplus.com/reference/stack/stack/uses_allocator/)

Uses allocator for stack (class template )

**Queue**

**Member functions**

[**(constructor)**](http://www.cplusplus.com/reference/queue/queue/queue/)

Construct queue (public member function )

[**empty**](http://www.cplusplus.com/reference/queue/queue/empty/)

Test whether container is empty (public member function )

[**size**](http://www.cplusplus.com/reference/queue/queue/size/)

Return size (public member function )

[**front**](http://www.cplusplus.com/reference/queue/queue/front/)

Access next element (public member function )

[**back**](http://www.cplusplus.com/reference/queue/queue/back/)

Access last element (public member function )

[**push**](http://www.cplusplus.com/reference/queue/queue/push/)

Insert element (public member function )

[**emplace**](http://www.cplusplus.com/reference/queue/queue/emplace/)

Construct and insert element (public member function )

[**pop**](http://www.cplusplus.com/reference/queue/queue/pop/)

Remove next element (public member function )

[**swap**](http://www.cplusplus.com/reference/queue/queue/swap/)

Swap contents (public member function )

**Non-member function overloads**

[**relational operators**](http://www.cplusplus.com/reference/queue/queue/operators/)

Relational operators for queue (function )

[**swap (queue)**](http://www.cplusplus.com/reference/queue/queue/swap-free/)

Exchange contents of queues (public member function )

**Non-member class specializations**

[**uses\_allocator<queue>**](http://www.cplusplus.com/reference/queue/queue/uses_allocator/)

Uses allocator for queue (class template )

**Deque**

**Member functions**

[**(constructor)**](http://www.cplusplus.com/reference/deque/deque/deque/)

Construct deque container (public member function )

[**(destructor)**](http://www.cplusplus.com/reference/deque/deque/~deque/)

Deque destructor (public member function )

[**operator=**](http://www.cplusplus.com/reference/deque/deque/operator=/)

Assign content (public member function )

**Iterators**:

[**begin**](http://www.cplusplus.com/reference/deque/deque/begin/)

Return iterator to beginning (public member function )

[**end**](http://www.cplusplus.com/reference/deque/deque/end/)

Return iterator to end (public member function )

[**rbegin**](http://www.cplusplus.com/reference/deque/deque/rbegin/)

Return reverse iterator to reverse beginning (public member function )

[**rend**](http://www.cplusplus.com/reference/deque/deque/rend/)

Return reverse iterator to reverse end (public member function )

[**cbegin**](http://www.cplusplus.com/reference/deque/deque/cbegin/)

Return const\_iterator to beginning (public member function )

[**cend**](http://www.cplusplus.com/reference/deque/deque/cend/)

Return const\_iterator to end (public member function )

[**crbegin**](http://www.cplusplus.com/reference/deque/deque/crbegin/)

Return const\_reverse\_iterator to reverse beginning (public member function )

[**crend**](http://www.cplusplus.com/reference/deque/deque/crend/)

Return const\_reverse\_iterator to reverse end (public member function )

**Capacity**:

[**size**](http://www.cplusplus.com/reference/deque/deque/size/)

Return size (public member function )

[**max\_size**](http://www.cplusplus.com/reference/deque/deque/max_size/)

Return maximum size (public member function )

[**resize**](http://www.cplusplus.com/reference/deque/deque/resize/)

Change size (public member function )

[**empty**](http://www.cplusplus.com/reference/deque/deque/empty/)

Test whether container is empty (public member function )

[**shrink\_to\_fit**](http://www.cplusplus.com/reference/deque/deque/shrink_to_fit/)

Shrink to fit (public member function )

**Element access**:

[**operator[]**](http://www.cplusplus.com/reference/deque/deque/operator%5b%5d/)

Access element (public member function )

[**at**](http://www.cplusplus.com/reference/deque/deque/at/)

Access element (public member function )

[**front**](http://www.cplusplus.com/reference/deque/deque/front/)

Access first element (public member function )

[**back**](http://www.cplusplus.com/reference/deque/deque/back/)

Access last element (public member function )

**Modifiers**:

[**assign**](http://www.cplusplus.com/reference/deque/deque/assign/)

Assign container content (public member function )

[**push\_back**](http://www.cplusplus.com/reference/deque/deque/push_back/)

Add element at the end (public member function )

[**push\_front**](http://www.cplusplus.com/reference/deque/deque/push_front/)

Insert element at beginning (public member function )

[**pop\_back**](http://www.cplusplus.com/reference/deque/deque/pop_back/)

Delete last element (public member function )

[**pop\_front**](http://www.cplusplus.com/reference/deque/deque/pop_front/)

Delete first element (public member function )

[**insert**](http://www.cplusplus.com/reference/deque/deque/insert/)

Insert elements (public member function )

[**erase**](http://www.cplusplus.com/reference/deque/deque/erase/)

Erase elements (public member function )

[**swap**](http://www.cplusplus.com/reference/deque/deque/swap/)

Swap content (public member function )

[**clear**](http://www.cplusplus.com/reference/deque/deque/clear/)

Clear content (public member function )

[**emplace**](http://www.cplusplus.com/reference/deque/deque/emplace/)

Construct and insert element (public member function )

[**emplace\_front**](http://www.cplusplus.com/reference/deque/deque/emplace_front/)

Construct and insert element at beginning (public member function )

[**emplace\_back**](http://www.cplusplus.com/reference/deque/deque/emplace_back/)

Construct and insert element at the end (public member function )

**Allocator**:

[**get\_allocator**](http://www.cplusplus.com/reference/deque/deque/get_allocator/)

Get allocator (public member function )

**Non-member functions overloads**

[**relational operators**](http://www.cplusplus.com/reference/deque/deque/operators/)

Relational operators for deque (function )

[**swap**](http://www.cplusplus.com/reference/deque/deque/swap-free/)

Exchanges the contents of two deque containers (function template )

**Mathematical Formulas**

Binomial:

Expansion:

Harmonic Series:

AP, nth term:

Sum to n terms:

Where L is the last term of the series of n terms

GP, nth term:

Sum to n terms:

Sum to infinity:

|  |
| --- |
| **Double Circular Linked List with dummy head implementation**  struct node {  char data;  struct node \*next;  struct node \*prev;  node (char data){ //Constructor used to add a new node  this->data = data; //Assign user input data to the node data (using "this" pointer)  next = NULL; //Intialize the new node to point to NULL  prev = NULL;  }  };  //Implmentation of a Double Circular Linked List with a dummy head  class linkedlist {  private:  node \*head; //Pointer to head of the list  node \*i; //Iterator to move through list  node \*dummy = new node ('\0'); //Dummy head  public:  linkedlist(){ //Constructor used to intialize empty linked list with dummy head  head = dummy;  dummy->next = dummy->prev = dummy;  i = dummy;  }  //Shift i to mark start of list  void begin(){  i = dummy->next; //i points to position after dummy head  }  //Shift i to mark end of list  void end(){  i = dummy; //i points to position after tail, which is the dummy head  }  //Erase the data at the position before i  void erase(){    if (i == dummy->next){  return; //If i points to start of list, do nothing  }  //Create a temp node to store the node to be erased  node \*temp;  temp = i->prev; //Note that i is pointing to position after node that will be erased  //Perform the actual deletion process here  temp->prev->next = i;  i->prev = temp->prev;  delete temp;    }  //Insert data at the position before the iterator i  void insert(char data){  node \*extra = new node (data);    i->prev->next = extra;  extra->prev = i->prev;  i->prev = extra;  extra->next = i;  }  //Clear the linked list of all data  void clear(){    while (dummy->next != dummy && dummy->prev != dummy){  node \*temp;  temp = dummy->next;  dummy->next = temp->next;  temp->next->prev = dummy;  delete temp;  }  }  void printlist(){    node \*temp = dummy->next;    while (temp != dummy){  cout << temp->data;  temp = temp->next;  }  }  }; |

|  |
| --- |
| **Single Linked List Implementation**  class SSL {  struct node{  V data;  struct node\* next;  };  private:  node\* head;  int len;  public:  SSL() {  head = NULL;  len = 0;  }  struct node\* createNode(V val){  node\* cur = new node();  cur->data = val;  cur->next == NULL;  return cur;  }  void insertFirst(V v){  len++;  node\* cur = createNode(v);  // if empty list, the 1st inserted node will be the head  if (head == NULL) {  head = cur;  return;  }  cur->next = head;  head = cur;  }  void removeFirst() {  if (head == NULL) {  return;  }  node\* preHead = head;  head = head->next;  len--;  delete preHead;  }  void reverseList(){  // no change if list has zero or one element  if(head == NULL || head->next == NULL)  return;  node\* cur = head->next;  head->next = NULL;  while(cur != NULL){  node\* temp = cur->next;  cur->next = head;  head = cur;  cur = temp;  }  }  void printList(){  node\* cur = head;  while(cur != NULL){  cout << cur->data << " ";  cur = cur->next;  }  cout << endl;  }  void insertAtIndex(V v, int pos) {  if (pos == 0) {  insertFirst(v);  return;  }  if (pos < 0 || pos > len) {  return;  }  len++;  node\* cur = createNode(v);  node\* pre = head;  while (--pos) {  pre = pre->next;  }  cur->next = pre->next;  pre->next = cur;  }  int findElement(V x) {  int pos = 0;  node\* cur = head;  while (cur) {  if (cur->data == x) {  return pos;  }  pos++;  cur = cur->next;  }  return -1;  }  int length() {  return len;  }  }; |