

Commonly Used C++ Built-In Functions

Sr#	Function	Header	Use	Example
1	sort(begin, end)	<algorithm>	Sorts a container	sort(vec.begin(), vec.end());
2	reverse(begin, end)	<algorithm>	Reverses range	reverse(vec.begin(), vec.end());
3	max(a, b)	<algorithm>	Returns max	int m = max(3, 5);
4	min(a, b)	<algorithm>	Returns min	int m = min(3, 5);
5	swap(a, b)	<algorithm>	Swaps values	swap(a, b);
6	find(begin, end, val)	<algorithm>	Finds value	find(vec.begin(), vec.end(), 3);
7	count(begin, end, val)	<algorithm>	Counts value	count(vec.begin(), vec.end(), 3);
8	accumulate(begin, end, init)	<numeric>	Sum of elements	accumulate(vec.begin(), vec.end(), 0);
9	lower_bound(begin, end, val)	<algorithm>	Lower bound	lower_bound(vec.begin(), vec.end(), 3);
10	upper_bound(begin, end, val)	<algorithm>	Upper bound	upper_bound(vec.begin(), vec.end(), 3);
11	binary_search(begin, end, val)	<algorithm>	Binary search	binary_search(vec.begin(), vec.end(), 3);
12	next_permutation(begin, end)	<algorithm>	Next permutation	next_permutation(vec.begin(), vec.end());
13	prev_permutation(begin, end)	<algorithm>	Previous permutation	prev_permutation(vec.begin(), vec.end());
14	to_string(int)	<string>	Int to string	string s = to_string(123);
15	stoi(string)	<string>	String to int	int x = stoi("123");
16	substr(pos, len)	<string>	Substring	str.substr(2, 3);
17	push_back(val)	<vector>	Append element	vec.push_back(5);
18	pop_back()	<vector>	Remove last	vec.pop_back();
19	emplace_back(val)	<vector>	Efficient insert	vec.emplace_back(5);
20	insert(pos, val)	<vector>	Insert at pos	vec.insert(vec.begin()+1, 5);
21	erase(pos)	<vector>	Erase at pos	vec.erase(vec.begin()+1);
22	unordered_map<k,v>	<unordered_map>	Hash map	unordered_map<int,int> mp;
23	set / unordered_set	<set>	Unique elements	set<int> s;
24	priority_queue	<queue>	Max-heap	priority_queue<int> pq;
25	queue / stack / deque	<queue>	Linear DS	stack<int> st;
26	make_pair(a, b)	<utility>	Make pair	auto p = make_pair(1, 2);
27	tie(a, b)	<tuple>	Unpack pair	tie(a, b) = p;
28	iota(begin, end, start)	<numeric>	Fill increasing	iota(vec.begin(), vec.end(), 0);