std::set_intersection in C++

The intersection of two sets is formed only by the elements that are present in both sets. The elements of two sets must be sorted before using the function

Syntax

```
OutputIterator set_intersection (  InputIterator1 first1, InputIterator1 last1,
                                   InputIterator2 first2, InputIterator2 last2,
                                   OutputIterator result);
```

first1, last1

Input iterators to the initial and final positions of the first sorted sequence. The range used is [first1, last1), which contains all the elements between first1 and last1, including the element pointed by first1 but not the element pointed by last1.

first2, last2

Input iterators to the initial and final positions of the second sorted sequence. The range used is [first2, last2).

result

Output iterator to the initial position of the range where the resulting sequence is stored.

The pointed type shall support being assigned the value of an element from the first range.

i.e,it points the beginning reference of the new set/vector of common elements

Return Type :

An iterator to the end of the constructed range.(it points reference to end of the common elements vector/sets)

```
// CPP program to illustrate
// std :: set_intersection

#include <bits/stdc++.h>

int main()
{
	int first[] = { 5, 10, 15, 20, 25 };
	int second[] = { 50, 40, 30, 20, 10 };
	int n = sizeofS(first) / sizeof(first[0]);

	std::vector<int> v1(5);
	std::vector<int> v2(5);
	std::vector<int>::iterator it, ls; // iterator of vector declaration

	std::sort(first, first + 5);
	std::sort(second, second + 5);

	// Print elements
	std::cout << "First array :";
	for (int i = 0; i < n; i++)
		std::cout << " " << first[i];
	std::cout << "\n";

	// Print elements
	std::cout << "Second array :";
	for (int i = 0; i < n; i++)
		std::cout << " " << second[i];
	std::cout << "\n\n";

	// std :: set_intersection
	ls = std::set_intersection(first, first + 5, second, second + 5, v1.begin());

	std::cout << "The intersection has " << (ls - v1.begin()) << " elements:";
	for (it = v1.begin(); it != ls; ++it)
		std::cout << ' ' << *it;
	std::cout << "\n";

	return 0;
}
```

output

First array : 5 10 15 20 25

Second array : 10 20 30 40 50

The intersection has 2 elements: 10 20