```
// C++ program to illustrate the
// capacity function in vector
#include <iostream>
#include <vector>

using namespace std;

int main()
{
	vector<int> g1;
vector<int>::iterator it;
	for (int i = 1; i <= 5; i++)
		g1.push_back(i);

	cout << "Size : " << g1.size();
	cout << "\nCapacity : " << g1.capacity();
	cout << "\nMax_Size : " << g1.max_size();

	// resizes the vector size to 4
	g1.resize(4);

	// prints the vector size after resize()
	cout << "\nSize : " << g1.size();

	// checks if the vector is empty or not
	if (g1.empty() == false)
		cout << "\nVector is not empty";
	else
		cout << "\nVector is empty";

	// Shrinks the vector
```

```
//		g1.shrink_to_fit();
		cout << "\nVector elements are: ";
		for (it = g1.begin(); it != g1.end(); it++)
			cout << *it << " ";

		return 0;
}


// C++ program to illustrate the
// element access in vector
#include <bits/stdc++.h>
using namespace std;

int main()
{
		vector<int> g1;

		for (int i = 1; i <= 10; i++)
			g1.push_back(i * 10);

		cout << "\nReference operator [g] : g1[2] = " << g1[2];

		cout << "\nat : g1.at(4) = " << g1.at(4);

		cout << "\nfront() : g1.front() = " << g1.front();

		cout << "\nback() : g1.back() = " << g1.back();

		// pointer to the first element
		int* pos = g1.data();
```

```
	cout << "\nThe first element is " << *pos;
	return 0;
}


// C++ program to illustrate the
// Modifiers in vector
#include <bits/stdc++.h>
#include <vector>
using namespace std;

int main()
{
	// Assign vector
	vector<int> v;

	// fill the vector with 10 five times
	v.assign(5, 10);

	cout << "The vector elements are: ";
	for (int i = 0; i < v.size(); i++)
		cout << v[i] << " ";

	// inserts 15 to the last position
	v.push_back(15);
	int n = v.size();
	cout << "\nThe last element is: " << v[n - 1];

	// removes last element
	v.pop_back();
```

```
// prints the vector
cout << "\nThe vector elements are: ";
for (int i = 0; i < v.size(); i++)
        cout << v[i] << " ";

// inserts 5 at the beginning
v.insert(v.begin(), 5);

cout << "\nThe first element is: " << v[0];

// removes the first element
v.erase(v.begin());

cout << "\nThe first element is: " << v[0];



// erases the vector
v.clear();
cout << "\nVector size after clear(): " << v.size();

// two vector to perform swap
vector<int> v1, v2;
v1.push_back(1);
v1.push_back(2);
v2.push_back(3);
v2.push_back(4);

cout << "\n\nVector 1: ";
for (int i = 0; i < v1.size(); i++)
```

```
        cout << v1[i] << " ";

    cout << "\nVector 2: ";
    for (int i = 0; i < v2.size(); i++)
        cout << v2[i] << " ";

    // Swaps v1 and v2
    v1.swap(v2);

    cout << "\nAfter Swap \nVector 1: ";
    for (int i = 0; i < v1.size(); i++)
        cout << v1[i] << " ";

    cout << "\nVector 2: ";
    for (int i = 0; i < v2.size(); i++)
        cout << v2[i] << " ";
}
```