Problem 1: Vector

We aim to implement a dynamic ADT list based on arrays called Vector similar to the C++ vector STL. The main characteristic of a vector is that we can add many elements to the vector without worrying about the maximum limit size of the array since it is incremented implicitly when needed to store all the needed elements. In the following table, we gave a brief description about some of vector STL operations.

- Write a C++ program that implements Vector with similar operations given in the below table respecting the complexities in it and respecting the main characteristic described above.

STL vector operations	Description	Complexity
vector<int> v;	Make an empty integer vector	O(1)
vector<int> v(n);	Make a vector with n elements.	O(n)
vector<int> v(n, value);	Make a vector with n elements, initialized to value.	O(n)
v.assign(n,value)	It assigns new value to the vector elements by replacing old ones.	O(n)
v.resize(n)	Resizes the container so that it contains 'n' elements.	O(n)
v.size();	Return current number of elements.	O(1)
v.capacity()	Returns the size of the storage space currently allocated to the vector expressed as number of elements.	O(1)
v.empty();	Return true if vector is empty.	O(1)
v.front();	Return the first element.	O(1)
v.back();	Return the last element.	O(1)
v.push_back(value);	Push the elements into a vector from the back.	O(1) (average)
v.insert(position, value);	Insert value at the position indexed by position.	O(n)
v.pop_back();	Remove value from end.	O(1)
v.erase(position);	Erase value indexed by position.	O(n)