Vector resize() function :-

The resize() function in C++ is a member function of the std::vector container that allows you to change the size of the vector. It modifies the vector to have a new size, either increasing or decreasing it.

Syntax :-

1. void resize(size_t n)

This changes the size of the vector to n. If n is less than the current size of the vector, it truncates the vector. If n is larger than the current size, it expands the vector, & the new elements are initialized with the default value (usually 0 or a default-constructed value, depending on the type of the elements in the vector).

2. void resize(size_t n, const T& value)

This changes the size of the vector to n. If n is larger than the current size, the new elements are initialized with the specified value.

Detailed Behavior:

1. **When Reducing the Size:**
    - If the new size is smaller than the current size, the vector is truncated to the new size. Elements beyond the new size are discarded.
2. **When Increasing the Size:**
    - If the new size is larger than the current size, the vector is resized, and new elements are added. If you don't provide a value to initialize the new elements, they are initialized to the default value for that type (for example, 0 for `int`, `nullptr` for pointer types, or an empty string for `std::string`).

Important Considerations:

- **Performance:** When expanding a vector, if the new size exceeds the current capacity of the vector, the vector may need to reallocate its memory. This may involve copying existing elements into a new memory location, which can incur some overhead.
- **Default Initialization:** If you don't provide a value when increasing the size, the new elements are default-constructed, which can lead to different behaviors depending on the type (e.g., zeros for `int`, `nullptr` for pointers, etc.).