**MAP IN C++**

**implementation :**

map<key type,value type> variable name;

unordered\_map<key type,value type> variable name;

|  |  |
| --- | --- |
| **FUNCTIONS** | **Description** |
| **Map.end()** | Returns an iterator to the theoretical element that follows the last element in the map. |
| Map.begin() | Returns an iterator to the first element of the map |
| Map.size() | Returns the number of elements in the map |
| Map.max\_size() | Returns the maximum number of elements that the map can hold. |
| Map.empty() | Returns whether the map is empty. |
| insert(pair<int, int>(1, 40)) | Adds a new element to the map |
| Map.erase(key) | Deletes the key -value from the map. |
| Map.erase(iterator1) | Deletes the key-value at the particular position |
| Map.erase(iterator1,iterator2) | Deletes the key-value from iterator 1 to iterator 2 |
| Map.count(key) | Returns the number of matches to element with key-value ‘g’ in the map |
| Map.rend() | Returns a reverse iterator pointing to the theoretical element right before the first key-value pair in the map(which is considered its reverse end). |
| Map.rbegin() | returns a reverse iterator which points to the last element of the map |
| Map.find(key) | Returns an iterator to the element with key-value ‘g’ in the map if found, else returns the iterator to end. |
| Map.crbegin() and map.crend() | crbegin() returns a constant reverse iterator referring to the last element in the map container. crend() returns a constant reverse iterator pointing to the theoretical element before the first element in the map. Ie(values cannot be changed view only) |
| map cbegin() and map.cend() | cbegin() returns a constant iterator referring to the first element in the map container. cend() returns a constant iterator pointing to the theoretical element that follows the last element in the multimap. Ie(values cannot be changed) |
| Map.upper\_bond(key) | returns an iterator pointing to the immediate next element just greater than k. If the key passed in the parameter exceeds the maximum key in the container, then the iterator returned points to the number of elements in the map container as key and element=0. |
| Map.lower\_bond(key) | returns an iterator pointing to the key in the map container which is equivalent to k passed in the parameter. In case k is not present in the map container, the function returns an iterator pointing to the immediate next element which is just greater than k. If the key passed in the parameter exceeds the maximum key in the container, then the returned iterator points to the number of elements in the map as key and element= any element. |
| map\_name.emplace\_hint(position, key, element) | Inserts the element at the given position,this does not affect the order since only the searching start from the position |
| Map.emplace(key,value) | Inserts the key and its element in the map container. |
| **=** eg- map[i]=10; | Used to assing he key value |
| [] map[i] | Used to access the vale |
| Map.at(key) | Same as [] operator |

Unordered\_map vs ordered\_map

