Member Functions of the map Class

constructors	Create maps
operator=	Copy the contents of a map
operator[]	Return a reference to the element of a map that is associated with a specific key
begin	Return the iterator pointing to the beginning of a map
clear	Erase all elements of a map
count	Count the number of elements of a map with a specific key
empty	Test whether a map is empty
end	Return the iterator pointing to the end of a map
equal_range	Find a range of a map containing all elements with a specific key
erase	Erase elements of a map
find	Find the element of a map with a specific key
insert	Insert elements into a map
key_comp	Compare two keys of a map for ordering
lower_bound	Find the first element of map whose key is not less than a specific key
max_size	Return the largest possible size of a map
rbegin	Return the reverse_iterator to the beginning of a reversed map
rend	Return the reverse_iterator pointing to the end of a reversed map
size	Return the size of a map
swap	Swap the contents of two maps
upper_bound	Find the first element of a map whose key is greater than a specific key
value_comp	Compare two values of a map for ordering

Function Prototypes

constructors	Create maps
	map (const C& cmp =less<K>) – construct an empty map, using cmp as the comparison object for the keys of the map, which can either be a class implementing a function call operator or a pointer to a function
	template <class II> map (II first, II last, const C& cmp =less<K>) – construct a map from a copy of the elements, starting from the element referred by the input iterator first to the element right before the one referred by the input iterator last, using cmp as the comparison object for the keys of the map, which can either be a class implementing a function call operator or a pointer to a function
	map (const map<K, T, C>& m) – construct a copy of the map m
destructor	Destroy a map
	~map () – deallocate all the storage capacity allocated by a map
operator=	Copy the contents of a map
	map<K, T, C>& operator= (const map<K, T, C>& m) – assign a copy of the map m to a map
operator[]	Return a reference to the element of a map that is associated with a specific key
	T& operator[] (const K& k) – return a reference to an element of a map, which is mapped by the key k, and if there is no match for k, then it inserts a new element in the map with the key k, and the new element is constructed by the default constructor of the map class
begin	Return the iterator pointing to the beginning of a map
	iterator begin () – return an iterator referring to the first element in a map
	const_iterator begin () const – const version of the iterator
clear	Erase all elements of a map
	void clear () – set a map content to an empty map
count	Count the number of elements of a map with a specific key
	size_type count (const K& k) const – search a map for an element with the key k and return the number of elements with that key, which is either 1 or 0
empty	Test whether a map is empty
	bool empty () const – return whether a map is empty
end	Return the iterator pointing to the end of a map
	iterator end () – return an iterator referring to the end of a map
	const_iterator end () const – const version of the iterator
equal_range	Find a range of a map containing all elements with a specific key
	pair<iterator, iterator> equal_range (const K& k) – return the bounds of a range that includes all the elements in a map whose key values compare equal to k, which include one element at most
	pair <const_iterator, const_iterator> equal_range (const K& k) const – const version of the function
erase	Erase elements of a map
	void erase (iterator i) – erase the element of a map at the position referred by the iterator i
	size_type erase (const K& k) – erase the element with the key k in a map and return the number of elements erased, which is 1 if the element exists; otherwise, it's 0
	void erase (iterator first, iterator last) – erase all the elements of a map between the positions referred by the iterators first and last

find	Find the element of a map with a specific key
	iterator find (const K& k) – return an iterator referring to the position of the element with key k in a map if it's found; otherwise, return the iterator end
	const_iterator (const K& k) const – const version of the function
insert	Insert elements into a map
	pair<iterator, bool> insert (const pair<const K, T>& x) – insert a copy of the element x into a map, but the element is not inserted if another element exists in the map with the same key value of x, and it returns a pair, whose first element is referring to either the newly inserted element or to the existing element, and its second element is set to true if the new element is inserted
	iterator insert (iterator i, const pair<const K, T>& x) – insert a copy of the element x into a map, where i is referring to the position of the first element to be compared
	template <class II> void insert (II first, II last) – insert a copy of the elements, between the elements referred by the input iterators first and last, into a map
key_comp	Compare two keys of a map for ordering
	key_compare key_comp () const – return a comparison object associated with a map, which can be used to compare the keys of two elements in the map, where key_compare is a member type of the class
lower_bound	Find the first element of map whose key is not less than a specific key
	iterator lower_bound (const K& k) – return an iterator referring to the first element of a map whose key does not compare less than the key k, using the map's comparison object
	const_iterator lower_bound (const K& k) const – const version of the function
max_size	Return the largest possible size of a map
	size_type max_size () const – return the maximum number of elements that a map can hold
rbegin	Return the reverse_iterator to the beginning of a reversed map
	reverse_iterator rbegin () – return a reverse iterator referring to the last element of a map
	const_reverse_iterator rbegin () const – const version of the reverse iterator
rend	Return the reverse_iterator pointing to the end of a reversed map
	reverse_iterator rend () – return a reverse iterator referring to the element right before the first element of a map
	const_reverse_iterator rend () const – const version of the reverse iterator
size	Return the size of a map
	size_type size () const – return the number of elements in a map
swap	Swap the contents of two maps
	void swap (map<K, T, C>& m) – swap the contents of a map with the map m
upper_bound	Find the first element of a map whose key is greater than a specific key
	iterator upper_bound (const K& k) – return an iterator referring to the first element of a map whose key compares greater than the key k, using the map's comparison object
	const_iterator upper_bound (const K& k) const – const version of the function
value_comp	Compare two values of a map for ordering
	value_compare value_comp () const – return a comparison object associated with a map, which can be used to compare the values of two elements in the map, where value_compare is a member type of the class