***str.append(str2);*** (also ‘+=’ operator) (**O(N)** where N is the size of the new string)

***str.insert(pos\_to\_begin,string\_to\_insert);*** *str.insert(8,"Geeks");*

*---------------------------------------------------------------------------------------------------------------------------*

***1. copy(strt\_iter1, end\_iter1, strt\_iter2);***

*copy(v1.begin(), v1.begin()+3, v2.begin());*

*copy(v1[2], v1[8], v2[0]);*

***2. copy\_n(strt\_iter1, num, strt\_iter2);***

*copy\_n(v1.begin(), 4, v3.begin());*

*---------------------------------------------------------------------------------------------------------------------------*

***str.replace(strt\_index, nb\_of\_chars, str2*);** replaces b characters from a index by str

*str6.replace(2, 7, "abcdefg");*

***swap(str1,str2);***

***str.substr(strt\_index, nb\_of\_chars);*** //returns a substring of b length starting from index a

*cout << str6.substr(7, 3) << endl;*

If second argument is not passed, string till end is taken as substring

*cout << str6.substr(7) << endl;*

*---------------------------------------------------------------------------------------------------------------------------*

***str.clear();*** (O(N))

***str.erase(strt,nb\_chars);*** (O(N) where N is the size of the new string)

*str6.erase(7, 4);*

*Or str6.erase(str6.begin() + 5, str6.end() - 3);*

***str.pop\_back();***

***empty()***

*---------------------------------------------------------------------------------------------------------------------------*

***str.find(str2);***

// find() returns index where pattern is found.

// If pattern is not there it returns predefined constant npos whose value is -1

*if (str6.find(str4) != string::npos)*

*cout << "str4 found in str6 at " << str6.find(str4) << " pos" << endl;*

*---------------------------------------------------------------------------------------------------------------------------*

***length();***

***str.resize(new\_size);***

---------------------------------------------------------------------------------------------------------------------------

\*Iterator Functions:

***str.begin();***

***str.end();***

***rbegin();*** This function returns a reverse iterator pointing at the end of string.

***rend();*** This function returns a reverse iterator pointing at beginning of string.