Thursday, 3 February, 2022 10:38 AM

String STL

- str.push_back(ch) : insert a character ch at the end of a string. TC-O(1)
- str.pop_back( ) : delete the last character of the string. TC-O(1)
- str.back( ) : return last character of a string. TC-O(1)
- str.front( ) : return the first element of the string. TC-O(1)
- swap(str[i] , str[j]) : swap characters at index i and j. TC-O(1)
- str.size() : return length of string. TC-O(1)
- toupper(ch) : converts a given character to uppercase.
- tolower(ch) : converts a given character to lowercase.
- string s =string(x,'*'); insert a character to a string x times.

Sort a string : TC-O(nlogn)

sort(str.begin(), str.end()) : sort string according to alphabetical order.
sort(str.rbegin(), str.rend()) : sort string in reverse of alphabetical order.

Reverse a string : TC-O(n)

reverse(s.begin(), s.end());	reverse whole string
reverse(s.begin(), s.begin()+x);	reverse first x characters
reverse(s.end()-x, s.end());	reverse last x characters

Question: **Reverse every word in given string. Ex- s="geeks for geek"**
output- skeeg rof keeg

str.erase( ): TC-O(n)

```
erase() :erase all string
s.erase ( pos) : erase all character after pos index.
erase ( idx, len ) :erase len character from idx index
erase (iterator pos) :erase character at pos index.
erase (iterator beg, iterator end ) : erase characters from beg to end.
```

```cpp
// string::erase
#include <iostream>
#include <string>

int main ()
{
  std::string str ("This is an example sentence.");
  std::cout << str << '\n';
                                       // "This is an example sentence."
  str.erase (10,8);                    //            ^^^^^^^^
  std::cout << str << '\n';
                                       // "This is an sentence."
  str.erase (str.begin()+9);           //           ^
  std::cout << str << '\n';
                                       // "This is a sentence."
  str.erase (str.begin()+5, str.end()-9);  //       ^^^^^
  std::cout << str << '\n';
                                       // "This sentence."
  return 0;
}
```

str.find( ): TC-O(n)

Function Template:

Default pos=0

size_t find (string str, size_t pos);

We can also search for a partial string

In below syntax, note that n is number of characters to match.

size_t find (string str, size_t pos, size_t n);

Function Return:

The function returns the index of the first occurrence of sub-string, if the sub-string is not found it returns string::npos(string::pos is static member with value as the highest possible for the size_t data structure)

```cpp
// string::find
#include <iostream>       // std::cout
#include <string>         // std::string

int main ()
{
  std::string str ("There are two needles in this haystack with needles.");
  std::string str2 ("needle");

  // different member versions of find in the same order as above:
  std::size_t found = str.find(str2);
  if (found!=std::string::npos)
    std::cout << "first 'needle' found at: " << found << '\n';

  found=str.find("needles are small",found+1,6);
  if (found!=std::string::npos)
    std::cout << "second 'needle' found at: " << found << '\n';

  found=str.find("haystack");
  if (found!=std::string::npos)
    std::cout << "'haystack' also found at: " << found << '\n';

  found=str.find('.');
  if (found!=std::string::npos)
    std::cout << "Period found at: " << found << '\n';

  // let's replace the first needle:
  str.replace(str.find(str2),str2.length(),"preposition");
  std::cout << str << '\n';

  return 0;
}
```

s.substr( ):

string s="abcdefg";
string s1=s.substr(1,3); s1="bcd"

Converting string to integer :

string s = "123";
   int num;

   // using stoi() to store the value of str1 to x
   num = stoi(s);

stof() - convert string to float
stod() - convert string to double
stold() - convert string to long double

stoll() - convert string to long long

Converting integer to string :

```
float numf = 123.4567;
int num = 123;

string str1 = to_string(numf);
string str2 = to_string(num);
```

Pairs

Pair is a container that stores two data elements in it. It is not necessary that the two values or data elements have to be of the same data type.

Syntax:

```
pair <datatype 1, datatype2 > pair_name;
```

The first value given in above pair could be accessed by using pair_name.first similarly, the second value could be accessed using pair_name.second.

```
//ex- pair<char,int> p;
```

Vector of Pairs

Vector of Pairs is a vector filled with pairs instead of any primitive data type.
To declare a vector of pairs we can use the syntax : *vector<pair<string,int>> a;*

To add a pair to an existing vector of pairs we can use the statement below:

```
v.push_back(make_pair("ABC",15));
v.push_back({"ABC",15});
```

push_back function helps in adding the elements to a vector, make_pair function converts the parameters into pairs.

https://tutorialspoint.dev/language/cpp/sorting-vector-of-pairs-in-c-set-1-sort-by-first-and-second