⇒ String: is a sequence of characters, terminated with a null character '\0'.

In c++, string is a sequence of characters as an object class. This class is called std::string

1. To read a string which contains whitespaces getline() function is used:

```
getline(cin, str, '\n');
```

The above function reads the input data from standard input and stored it in the 'str', until a new line has been entered.

2. stringstream(): associates a string object with a stream allowing you to read from the string as if it were a stream (like cin).

✗ count the number of words in a string:

```
int countwords(string str)
{
    stringstream s(str);
    string word;
```

```
int count = 0;
while (s >> word)
    count++;
return count;
}
```

3. String concatenation: Using '+' operator we can concatenate two strings.

```
String s1 = "fam";
String s2 = "ily";
cout << s1+s2 <<endl;
```

4. compare(): str.compare(str2) returns zero if both the strings are equal.

5. clear(): clear the contents of a string.

6. empty(): str.empty() checks whether a string is empty.

X. ASCII value of 'a': 97
'A': 65

- 'a' - 'A': 32 (The difference between the upper case & lower case letters).

* Using above logic we can convert the given string from lower case to upper case & vice-versa.

code :

```
for(int i=0; i<s.size(); i++)
{
    if (s[i]>='a' && s[i]<='z')
        s[i] = s[i] - 32;
}
```

* The Above code converts the given string from lower case to upper case. Similarly, the code given below converts the string from upper case to lower case.

```
for(int i=0; i<s.size(); i++)
{
    if(s[i]>='A' && s[i]<='Z')
        s[i] += 32;
}
```

=> Using transform() which is a built-in method.

```
transform(s1.begin(), s1.end(), s1.begin(),
                                    ::toupper);
```

Range of elements.

⇒ Biggest or largest possible Number from a numeric string:

Given: String s1 = "76899";
we have to return: "99876";

so, the approach here is to sort the string in decending order (Decreasing order to the characters)

```
s1.sort(s1.begin(), s1.end(), greater<int>());
```

⇒ max frequency of a character in a given string:

First, we will declare an array of size 26 (26 characters in english). will set all 26 locations to 0. we assume that the given string contains only lower case letters

```
for(int i=0; i<s.size(); i++)
{
    freq[s[i]-'a']++;
}
```

ex: s[i] = 'a', so, freq['a'-'a'] = freq[0]++

⇒ std::prev(); returns a iterator pointing to the element after being advanced by certain no. of positions in the reverse direction.

ex: To remove a character from the string.
last

> s.erase(prev(s.end()));