Conversion of String to char array and char array to String in C++

❖ Convert String to char array

1. The c_str() and strcpy() function in C++

C++ c_str() function along with C++ String strcpy() function can be used to convert a string to char array easily.

The c_str() method represents the sequence of characters in an array of string followed by a null character (‘\0’). It returns a null pointer to the string.

Syntax:

string-name.c_str();

Algorithm:

- At first, we use c_str() method to get all the characters of the string along with a terminating null character.
- Further, we declare an empty array of type char to store the result i.e. result of the conversion of string to char array.
- Finally, we use strcpy() method to copy the character sequence generated by the c_str() method to the empty char array.

Example:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
        string str = "";
        cout<<"Enter the string:\n";
        //cin>>str;
```

```
        getline (cin, str);
        char arr[str.length() + 1];

        strcpy(arr, str.c_str());
   cout<<"String to char array conversion:\n";
        for (int i = 0; i < str.length(); i++)
                cout << arr[i];

        return 0;
}
```

2. String to Char Array Conversion in C++ Using for Loop in

For the conversion of char array to a string, we can use C++ for loops with ease.

- Initially, we create an empty array of type char
- After this, we iterate through the input string
- While iterating, we store the characters into the char array.

Example:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
        string str = "";
        cout<<"Enter the string:\n";

        getline (cin, str);
   cout<<str;
        char arr[str.length() + 1];
   cout<<"String to char array conversion:\n";
   for (int x = 0; x < sizeof(arr); x++) {
      arr[x] = str[x];
```

```
    cout << arr[x];
  }
      return 0;
}
```

❖ Convert Char Array to String in C++

The mentioned techniques can be used to convert a char array to string in C++:

1. The ‘+’ operator
2. C++ overloaded ‘=’ operator
3. C++ inbuilt-constructor

1. The ‘+’ operator

C++ provides us with '+' operator to concatenate or add data items to a variable.

We create a new empty string to store the result.
Taking it ahead, we use a for loop to traverse through the input char array.
In the process of traversing through the array, we use ‘+’ operator to concatenate the characters to the string.

Example:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
        char arr[] = { 'H', 'i',' ' ,'H', 'e', 'l', 'l', 'o', 'C', 'l', 'a', 's', 's' };

        int size_arr = sizeof(arr) / sizeof(char);
        string str = "";
        for (int x = 0; x < size_arr; x++) {
                str = str + arr[x];
        }
```

```
	cout<<"Converted char array to string:\n";
	cout << str << endl;
	return 0;
}
```

2. C++ overloaded ‘=’ operator

C++ has got the concept of overloading which makes an operator perform other operations apart from the basic or default operation.

Initially, we create a new empty string.

We use the overloaded '=' operator to store the data items character by character into the newly created empty string.

Example:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
	char arr[] = { 'H', 'i',' ' ,'H', 'e', 'l', 'l', 'o', 'C', 'l', 'a', 's', 's' };
	int size_arr = sizeof(arr) / sizeof(char);
	string str = "";
	str = arr;
	cout<<"Converted char array to string:\n";
	cout << str << endl;
	return 0;
}
```

3.C++ String inbuilt constructor

In the context of conversion of char array to string, we can use C++ String Constructor for the same.

Syntax:
string string-name(char array-name);

This constructor takes a sequence of characters terminated by a null character as an input parameter.
Note: This constructor string string() can be used only at the time of string declaration throughout the program.

Example:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
	char arr[] = { 'H', 'i',' ' ,'H', 'e', 'l', 'l', 'o', 'C', 'l', 'a', 's', 's' };
	int size_arr = sizeof(arr) / sizeof(char);
	string str(arr);
	cout<<"Converted char array to string:\n";
	cout <<str<< endl;
	return 0;
}
```