USACO Bronze Basic A - 9. Strings

1. Strings

- The class provided by the C++ STL, literally a class that handles strings.
- In C, strings are handled in the form of char[], but in C++, strings are used as a type of variable, making it much more versatile and easier to handle strings.
- Unlike char[], the '\0' character is not inserted at the end of the string, and the length of the string can be changed dynamically.

2. I/O of Strings

- Input/output is possible with the C++ input/output method, cin and cout, and the getline function can also be used, so we need to include header file name iostream
- scanf and printf in C cannot be used.

```
string str; // create string
cin >> str; // Receives the string before the space.
getline(cin, str); // The string up to '\n', that is, the entire line is input. (with spaces)
getline(cin, str, 'a') // Gets the string up to 'a' character.
getline(cin, str, '\n') //== getline(cin, str)
cout << str; // print a string
```

```
string str;
getline(cin, str);
cout << "str : " << str << '\n';
getline(cin, str, 'd');
cout << "str : " << str << '\n';
cin >> str;
cout << "str : " << str << '\n';
```

3. Initialize Strings

- We have to include header file name string like #include<string>

```
string str;
```

```
string str = "abcdef";
```

```
string str;
str = "abcdef";
```

```
string str("abcdef");
```

```
string str2(str1);
```

4. Operation for Strings

- Operators such as <, >, ==, and + can be used with the string class.
- String comparison (<, >, ==): You can compare the dictionary order of two strings or check whether they are identical.
- String concatenation (+): Connects two strings.

5. Member functions for Strings

```
str.at(index)
str[index]
str.front()
str.back()
```

```
str.size()
str.capacity()
str.resize(n)
str.resize(n, 'a')
str.shrink_to_fit()
str.reserve(n)
str.empty()
```

```
str.append(str2)
str.append(str2, n, m)
str.append(n, 'a')
str.insert(n, str2)
str.replace(n, k, str2)
str.clear()
str.erase(n, m)
str.erase(n, m) (iterator)
str.erase()
str.push_back(c)
str.pop_back()
str.assign(str2)
```

```
str.find("abcd")
str.find("abcd", n)
str.substr()
str.substr(n)
str.substr(n, k)
str.compare(str2)
swap(str1, str2)
```