# String as Singly Linked List

In many programming languages, the **String** class is typically implemented as an array of characters. Some other languages, such as Haskell, Erlang, and a few other functional programming languages, however, implement strings as **Singly Linked List (SLL)** of characters. Compared with the array implementations, the SLL ones are very easy for pattern matching because of the recursive structure of linked list. Let's don't worry about recursion in this assignment.

In this assignment, you will implement a **SLLString** class using SLL. Your **SLLString** class will include the following members:

- **SLLString();** //Default constructor
- **SLLString(const string& other);** //copy constructor taking a C++ string as parameter.
- **~SLLString();** // destructor.
- **SLLString(const SLLString& other);** //copy constructor taking another SLLString
- **SLLString& operator=(const SLLString& other);** // assignment constructor
- **int length();** // get length of this string.
- **SLLString& operator+= (const SLLString& other);** // concatenation
- **char& operator[](const int n);** //get character at index n.
- **int findSubstring(const SLLString& substring);** // find the index of the first occurrence of substring in the current string. Returns -1 if not found.
- **void erase(char c);** //erase all occurrences of character c from the current string.
- support **cout <<** operator.
- **Node* head;** // the head pointer to the SLL.

Each **Node** object should contain a **char** data type and a **next** pointer to Node.

Here are the sample usages:

```
int main(){

    SLLString str("Hello world!");

    SLLString newStr = new SLLString;

    newStr = str;

    newStr += SLLString(" CS@BC");

    newStr[6] = 'W';

    cout << newStr << endl; // Hello World! CS@BC

    cout << newStr.length() << endl; //18
```

```
    int loc = newStr.findSubstring("World");

    cout << loc << endl; // 6

    newStr.erase('l'); //erase the letter l.

    cout << newStr << endl; // Heo Word! CS@BC

    newStr.erase('C');

    cout << newStr << endl; // Heo Word! S@B

}
```