CS 111

string functions

Recall from C-strings slides

- Strings can be treated as character arrays
  - string s = “hello”;
  - char t[] = “hello”;
  - s and t are equivalent
- Because strings are stored as character arrays, we can reference characters by their position in the array
  - string name = “QC”;
  - cout << name[1]; // prints ‘C’

String member functions

- These functions are called on individual strings via the method selection operator (“.”)
- A few example functions that you should know:
  - length()
  - insert()
  - substr()
  - find()
  - rfind()

length()

- Returns the number of characters in a given string
- This is equivalent to size()

```
string animal = “cat”;
cout << animal.size(); // outputs 3
cout << animal.length(); // outputs 3
```

insert()

- Consider two strings, str1 and str2, with pos being a position in str1
- Inserts a string inside another string at a specified location

```
string str1 = “This is fun!”;
string str2 = “a lot of “;
// insert(position, string)
str1.insert(8, str2);
cout << str1; // outputs “This is a lot of fun!”
```

substr() – from specified position

- Extracts a substring starting at a specified position
- Given a string and a position
- substr(position)

```
string example = “Queens College”;
string substr1 = example.substr(7);
cout << substr1; // prints College
```

substr() – specific length, from specified position

- Extracts a substring of a specified length (number of characters) from a string, starting at a specified position
- Given a string, a position and a length of the substring
- substr(position, length)

```
string example = “Queens College”;
string substr1 = example.substr(0, 6);
cout << substr1; // prints Queens
```

find()

- Locates the index position in a string of the first occurrence of a specified string starting from the left
- If it does not find the specified string it returns -1

```
string name = “Queens College”;
int pos = name.find(“ege”);
cout << pos; // prints 11
```

rfind()

- Locates the index position in a string of the first occurrence of a specified string starting from the right
- If it does not find the specified string it returns -1

```
string name = “Queens”;
int pos = name.rfind(“e”);
cout << pos; // prints 3
```

Concatenation

- A simple way to combine two strings is to use the “+” operator

```
string firstName = “Queens”;
string lastName = “College”;
cout << firstName + lastName; // prints QueensCollege
```