CS 111

c-strings and characters

C-strings

- Examples of strings:
    - string name = “Queens”;
    - cout << “Hello World”;
- It turns out that strings are stored as character arrays, which are called c-strings.
    - char t[] = “hello”;

Storing text in C++

- Consider these two variables:
  - string s = “hello”;
  - char t[] = “hello”;

Sample code

```
#include <iostream>
using namespace std;

int main() {
    string s = "hello";
    char t[] = "hello";
    if(s == t) cout << "s and t are equal";
    else cout << "s and t are not equal";
    return 0;
}
```

Strings as character arrays

- Because strings are stored as character arrays, we can reference letters by their positions in the array
  - string name = “CUNY”;
  - cout << name[2]; // this will print the letter ‘N’

String methods

- Strings have several pre-defined library functions available
  - length() and size() return the number of characters in a string
  - string s = “city”;
  - cout << s.size(); // prints 4
- Notice the “.” operator usage
- We will go over several additional string functions next lab

ASCII Characters

- ASCII is a character encoding standard, where characters are associated with numbers
- See, for example:

http://www.asciitable.com

Character arithmetic

- You can cast an integer as a char

```
int x = 65;
cout << (char)x; // prints ‘A’ (ASCII 65)
cout << (char)(x+1);  // prints ‘B’ (ASCII 66)

int y = 97;
cout << (char)y; // prints ‘a’ (ASCII 97)
cout << (char)(y-32);  // prints ‘A’ (ASCII 65)
```

Sample code

```
#include <iostream>
using namespace std;

int main() {
    int x = 65;
    cout << (char)x << endl; // prints ‘A’ (ASCII 65)
    cout << (char)(x+1) << endl; // prints ‘B’ (ASCII 66)
    int y = 97;
    cout << (char)y << endl; // prints ‘a’ (ASCII 97)
    cout << (char)(y-32) << endl; // prints ‘A’ (ASCII 65)
    return 0;
}
```