STRINGS

Problem Solving with Computers-I

C-Strings

Q1: How are ordinary arrays of characters and C-strings similar and how are they dissimilar?

Which of the following is not a C string?

```
A. char mystr[5] =  "John";
B. char mystr[] = "Mary";
C. const char *mystr = "Jill";
D. char mystr[4]= {'J', 'i', 'l', 'l'};
```

Q2: Which of the following statements about the given code is FALSE?

```
char s1[5] = "Mark", s2[5] = "Jill";
for (int i = 0; i <= 5; i++)
    s1[i] = s2[i];
if (s1 != s2) cout<<"Strings are not equal"
s1 = "Art";
```

A. There is an out of bound access in the for loop

B. The entire for loop can be replaced by s1 = s2;

C. Cannot compare two C-strings with != or == operators

D. The last assignment statement is incorrect

C String Standard Functions #include <cstring>

```cpp
char s1[5] = "Mark", s2[5] = "Jill";
for (int i = 0; i <= 5; i++)
    s1[i] = s2[i];
if (s1 != s2) s1 = "Art";
```

- **`int strlen(char *string);`**
  - Returns the length not counting of `string` the null terminator
- **`int strcmp(char *str1, char *str2);`**
  - return 0 if `str1` and `str2` are identical (how is this different from `str1 == str2`?)
- **`int strcpy(char *dst, char *src);`**
  - copy the contents of string `src` to the memory at `dst`. The caller must ensure that `dst` has enough memory to hold the data to be copied.
- **`char* strcat(char *s1, char *s2);`**
  - concatenate the contents of string `s2` to `s2` and returns pointer to resulting string

Q3: What is the output of the following code?

```cpp
char s1[4] = "apple";
char s2[4] = "pineapple";
if (strcmp(s1, s2))
    cout << "Strings are not equal";
```

C strings vs. String class: What is the output of the code?

```
string s1 = "Mark";
string s2 = "Jill";
for (int i = 0; i <= s1.length(); i++)
    s2[i] = s1[i];
if (s1 == s2) s1 = "Art";
cout<<s1<<" "<<s2<<endl;
```

A. Mark Jill
B. Mark Mark
C. Art Mark
D. Compiler error
E. Run-time error

The C++ string class methods

```
string fruit = "Apple";
int len = fruit.length();
int pos= fruit.find('l');
string part= fruit.substr(1,3);
fruit.erase(2,3);
fruit.insert(2,"ricot");
fruit.replace(2,5,"ple");

Check out ctype for checks and conversions on characters
fruit[0]= tolower(fruit[0]);
isalpha(fruit[0])
```
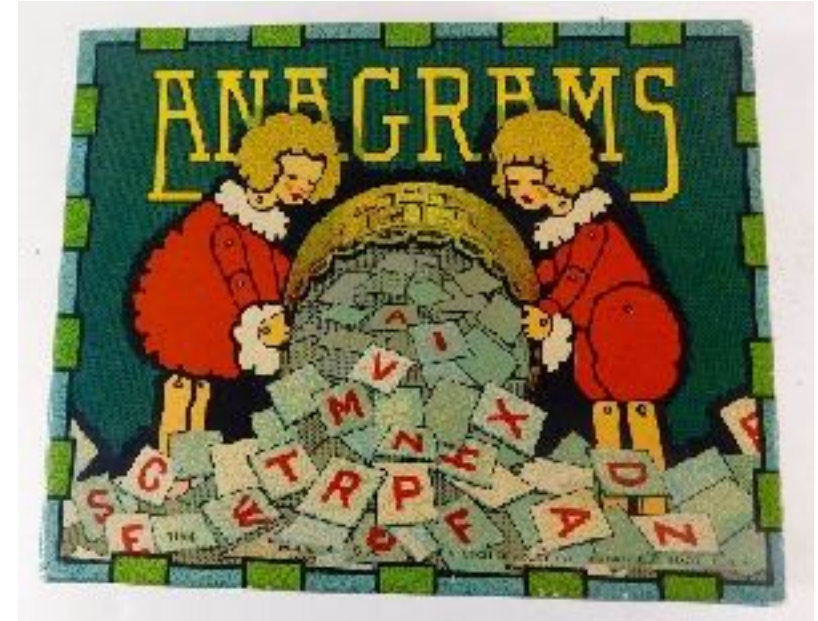
Lab 08: anagrams and palindromes

```
bool isAnagram(string s1, string s2)
```

Rats and Mice == In cat's dream
Waitress == A stew, Sir?

Why don't we pass the length of the string?

Next time

- Recursion