Lab 10: Character Strings

In C++ there are two ways to store sequences of characters ("character strings"): as an array of chars or as an object of the type 'string'. For both types of character strings there are a number of useful functions.

Arrays of characters

Arrays of characters are just like any other array. They are declared with commands like:

```cpp
char aString[10];
```

Elements of the array (that is any individual character in the array) can be accessed using its index just like any other array. A number of functions that are used to manipulate arrays of characters are defined in the header file `cstring`, which must be included to use them. Because C++ doesn’t keep track of the lengths of arrays a special character is placed at the end of character arrays so the string manipulation functions know when they’ve reached the end of a string. The character is called the NUL character and is denoted by '\0'.

String Objects

To use the string type you need to include the library called, simply enough, `string`. Object variables are declared just like variables of any basic data type:

```cpp
string anotherString;
```

Many of the function that apply to variables of the type string are accessed using operators. For example, the ‘+’ operator is used to concatenate two strings. Other functions in the string library are defined in the textbook.

In this lab you will be experimenting with different functions for both types of character strings: arrays of chars (C strings) and string objects.

Exercise 10.1

Write short pieces of code that demonstrate how each of the following functions work. Both the code and the output should clearly show what the functions do. For example, the following code could be used to demonstrate the `strlen()` function:
#include <iostream>
#include <cstring>

using namespace std;

int main()
{
    char test_string[] = "Test String";
    cout << "The strlen() function returns ";
    cout << "the length of a string" << endl;
    cout << "E.g. the length of the string: *";
    cout << test_string << "* is " << strlen(test_string);
    cout << "." << endl;

    return 0;
}

(The *'s are put in to make it clear where the string begins and ends.)

Turn in your code and the output demonstrating the function’s use.

Demonstrate the following functions from the cstring library that apply to character arrays:

1. strlen
2. strcpy
3. strcat
4. strcmp
5. strstr

Demonstrate the following functions from the string class library that apply to objects (variables) of type string:

1. length
2. = (string copy)
3. + (string concatenation)
4. == (string comparison)
5. find