Lab Exercises wk11 – Practice with Strings

Required Reading

None

Instructions – PLEASE READ *(notice **bold** and underlined phrases)*

Lab Exercise has three parts:

A. Lab Demo – Review of Strings
B. Exercises – Work Assignment 3, Problems 3B a,b,c, and h
C. Submission – Submit your solutions to

1. **You are to work on these lab exercises as individuals. Each student must reproduce the demo and show the TA that they have completed Part A.**
2. **Submission deadline: Friday Mar 18th at 10:30am**
3. **The exercises are presented in sequence so that you gradually advance with the material.**
4. **Before you leave the CSIL labs, make sure that a TA looks at your work in order to receive your attendance and lab active participation marks.**
5. **Lab11 Intended learning outcomes**

By the completion of the demo, students should be able to use NetBeans to:

- Be familiar with how C Strings are declared, stored, and manipulated
- Review how C++ Strings are declared, stored, and manipulated
- Know how to convert C Strings to C++ Strings, and C++ Strings to C Strings

A. Lab Demo – Presented by Scott

Your lecturer will present an overview of C Strings and demonstrate how to do the first question from Assignment #3 Problem B3.

.

3. Practice with Dynamic Arrays as Return Values

(a) helloName

Given a C string parameter called name, e.g. "Bob", return a greeting of the form "Hello Bob!".

Remember that C strings must be null terminated, so the above string would contain the four chars: name[0]='B', name[1]='o'; name[2]='b'; name[3]=0; Your function will accept a C string that is null terminated, and must return a new dynamic char array that is also null terminated.

Your function must have the following signature:

```
char* helloName(const char name[]);
```

For example:

```
helloName("Bob")     → "Hello Bob!"
helloName("Alice")   → "Hello Alice!"
helloName("X")       → "Hello X!"
helloName(nullptr)   → "Hello !"
```

B. Lab Exercises – To be completed by Students Individually

You must complete and submit solutions to the first three programming problems from Assignment 3 Problem B3: a, b, c, and h.

3. Practice with Dynamic Arrays as Return Values

(a) helloName

Given a C string parameter called `name`, e.g. "Bob", return a greeting of the form "Hello Bob!".

Remember that C strings must be null terminated, so the above string would contain the four chars: name[0]='B', name[1]='o'; name[2]='b'; name[3]=0; Your function will accept a C string that is null terminated, and must return a new dynamic char array that is also null terminated.

Your function must have the following signature:

```
char* helloName(const char name[]);
```

For example:

```
helloName("Bob")     → "Hello Bob!"
helloName("Alice")   → "Hello Alice!"
helloName("X")       → "Hello X!"
helloName(nullptr)   → "Hello !"
```

(b) makeAbba

Given two C strings, a and b, return the result of putting them together in the order abba, e.g. "Hi" and "Bye" returns "HiByeByeHi".

Your function must have the following signature:

```
char* makeAbba (const char a[], const char b[]);
```

For example:

```
makeAbba("Hi", "Bye")     → "HiByeByeHi"
makeAbba("Yo", "Alice")   → "YoAliceAliceYo"
makeAbba("What", "Up")    → "WhatUpUpWhat"
```

(c) doubleChar

Given a string, return a C string where for every char in the original, there are two chars.

Your function must have the following signature:

```
char* doubleChar (const char str[]);
```

For example:

```
doubleChar("The")         → "TThhee"
doubleChar("AAbb")        → "AAAAbbbb"
doubleChar("Hi-There")    → "HHii--TThheerree"
```

(h) fizzArray2

Given non-negative number n, return a new array of strings of length n, containing the C++ strings "0", "1" "2"... n-1. If n is 0 or less, return `nullptr`. Note that to_string() constructs string for most numeric types.

The syntax to make a new string array is: new string[desired_length]

Your function must have the following signature:

```
   string* fizzArray2 (int n);
```

For example:

```
   fizzArray2(4) → {"0", "1", "2", "3"}
   fizzArray2(2) → {"0", "1"}
   fizzArray2(0) → nullptr
```

C. Lab Exercise Submission – To be completed by Students

Students are responsible for submitting the requested work files by the stated deadline for full marks. Since Lab Exercise solutions may be discussed in class following the submission deadline, late submissions will NOT be accepted. It is the student's responsibility to submit on time. If you do not have access to CSIL or issues with the computers in CSIL, please contact the CSIL Help Desk at `helpdesk@cs.sfu.ca`

Students must work on these exercises individually and submit the set of files to CourSys. No group should be created for this submission.

1. **You must submit your final version of the following file before the deadline. Students must ensure that all submitted code compiles and is properly commented and formatted for readability:**
   - `helloName.cpp`
   - `makeAbba.cpp`
   - `doubleChar.cpp`
   - `fizzArray2.cpp`
2. **Files are to be submitted into CourSys under Lab11.**