CSE 1321L: Programming and Problem Solving I Lab

Lab 11

String Libraries

What students will learn:

- How to use functions that take strings as inputs

Overview: Strings are very different from the primitive data types we have studied thus far. Typically when comparing 2 variables, we can use basic comparisons tools such as <, >, ==, or !=. When dealing with a number, it is very easy to understand what this means. Once you understand chars can be converted to numbers in ASCII, this too is easy to compare. However strings are more complex! They contain a collection of chars which makes it a little more difficult to compare their various properties. Because of the widespread use of strings as a data type, there are pre-made functions which solve this problem and many others.

Check out the links below for documentation on how your language makes a string.

C++
https://en.cppreference.com/w/cpp/string/basic_string

C#
https://docs.microsoft.com/en-us/dotnet/api/system.string?view=net-5.0

Java
https://docs.oracle.com/javase/7/docs/api/java/lang/String.html

You may notice that a string is really just a group of chars organized in an array. You should feel comfortable with numerical arrays at this point. You can use that knowledge to help you understand how this structure would look for a string.

Index	0	1	2	3	4	5	6
Element	S	t	r	i	n	g	s
ASCII	83	116	114	105	110	103	115

This diagram above depicts how strings are read. There are a number of “cells” in this structure, and they are filled with an element which is the char or number at a particular block. They are ordered, starting at 0, numerically and can be referenced by an index number. By creating this structure of chars, when you want to compare two different letters it becomes a little more clear how to do this.

https://www.ascii-code.com/ (use the “DEC” numbers)

As always, don’t forget to include your name/section as comments for the header, document your code (also with comments) and call the class names and filenames the correct things (e.g. Lab11A .java, .cs, .cpp).

Lab11A: OK vs ok

You have probably gotten an automated text where it says something like “Text STOP to opt out of getting notifications”. If this happens do you reply with “STOP”, “Stop”, or “stop”? All 3 of these are completely different strings. How should you develop an application which would allow a user to input any of them?

For this first problem, create a very basic menu which displays a few different strings. Then after the program has finished, it will ask the user if they want to rerun the program. The user should be able to type “Yes”, “yes”, “YES”, or any other combination of lowercase and uppercase letters in the word.

DO NOT solve this problem using a complicated IF statement which will take all these inputs. Also, assume if the user types anything other than “yes” then the user does not want to rerun the program.

Tip: You need to use a function to compare if 2 strings are equal to each other. Try searching something like “<your language> string function”, there are many different functions available.

Remember, the class name should be Lab11A.
The user input is indicated in bold.

Sample output #1:

```
Menu
0) Hello World
1) Goodbye Moon
2) Walking on Sunshine

What would you like to do: 0

Hello!

Type YES to rerun. yes

Menu
0) Hello World
1) Goodbye Moon
2) Walking on Sunshine

What would you like to do: 1

Ok, bye.

Type YES to rerun. YeS

Menu
0) Hello World
1) Goodbye Moon
2) Walking on Sunshine

What would you like to do: 2

WHOA!

Type YES to rerun. YES

Menu
0) Hello World
1) Goodbye Moon
```

```
2) Walking on Sunshine

What would you like to do: 2

WHOA!

Type YES to rerun. No

Program Ends
```

Sample output #2:

```
Menu
0) Hello World
1) Goodbye Moon
2) Walking on Sunshine

What would you like to do: 0

Hello!

Type YES to rerun. yEs


Menu
0) Hello World
1) Goodbye Moon
2) Walking on Sunshine

What would you like to do: 0

Hello!

Type YES to rerun. No way!

Program Ends
```

Intentionally Left Empty, please proceed to Next Page

Lab11B: Password

This next problem is going to deal with checking if someone inputs a string with a few requirements. Imagine you are prompted to input a password, but the password needs to include uppercase letters and a number. In order to do this, you must look at each char in the string and use Booleans to indicate certain criteria are met. Since there are 3 criteria, you should have 3 Boolean variables.

The rules for the password are:

- Must contain at least 8 chars
- Must contain 1 uppercase letter
- Must contain 1 digit
- There are no restrictions involving lowercase letters or special chars

When considering where to use Booleans, think of it as a "flag", for each criteria. If you meet the length requirement, then the flag would change from FALSE to TRUE. Once all the flags are true, you will have a valid password.

Tip: The ASCII table can be used to determine the numeric value of a particular char. You may want to create ranges of these numerical values for each criteria.

```
Sample output #1:
Enter a password: password1
Invalid password

Program Ends

Sample output #2:
Enter a password: P4ssw3rd
Valid password

Program Ends

Sample output #3:
Enter a password: Pas1
Invalid password

Program Ends

Sample output #4:
Enter a password: PASSword
Invalid password

Program Ends

Sample output #5:
Enter a password: Pas!w3rd
Valid password

Program Ends
```

Instructions:

- Programs must be working correctly.
- Programs must be saved in files with the correct file name.
- If working in Java or C#, class names must be correct.
- Programs (only .java, .cs or .cpp files) must be uploaded to Gradescope by due date.