CS1050 – Prelab 8
Fall 2019

Concepts to Practice

- Characters
- Strings
- Standard Library functions for characters

Description

For the prelab assignment, you need to implement a program that takes pre-defined strings and categorizes those strings as all-digits, all-lower-case, all-upper-case, or mixed.

The main() function in your program is provided below, and should be the only code in main.c. You should create a separate file called stringfunc.c and compile the two together as shown in the example in order to test your code. Note that you should include the correct header file(s) in stringfunc.c so you will have the prototype(s) you need. You may also need to have prototypes for the function(s) you write in stringfunc.c as well.

You can use any of the Standard C Library functions that takes a character as an argument. You can also use printf. You should not need any other Standard C Library functions.

Functions You Must Write

You may write any functions you wish to implement this program, in **addition** to the following function. However, you **must** implement the following function, and it must be prototyped as shown:

- **void PrintCategory(char *s)** – This function takes a string and prints that string along with which category the string belongs to. See the sample output.

Main.c

Use this code for your main.c file:

```
// Symbolic Constants
#define DIGITTEST1 "12345abcd67890"
#define DIGITTEST2 "1234567890"
#define LOWERTEST1 "hifromJimRies"
#define LOWERTEST2 "hifromjimries"
#define UPPERTEST1 "JIMRIESWASHERE"
#define UPPERTEST2 "JIMRIESWASHERe"

// Prototypes
void PrintCategory(char *s);

// Main
int main(void)
{
	PrintCategory(DIGITTEST1);
	PrintCategory(DIGITTEST2);
	PrintCategory(LOWERTEST1);
	PrintCategory(LOWERTEST2);
	PrintCategory(UPPERTEST1);
	PrintCategory(UPPERTEST2);
}
```

How to Compile and Link

When you have coded main.c (as shown above) and created stringfunc.c, compile the two as follows:

compile main.c stringfunc.c

At this point (if you have no errors), you can run the program as usual:

./a.out

Sample Output

```
jimr@JimRHadesCanyon:$ compile main.c stringfunc.c
jimr@JimRHadesCanyon:$ ./a.out
12345abcd67890 is a mix of various types of characters.
1234567890 is all digits.
hifromJimRies is a mix of various types of characters.
hifromjimries is all lower case.
JIMRIESWASHERE is all upper case.
JIMRIESWASHERe is a mix of various types of characters.
```