CH10_006

EXERCISE

USING THE TORQUE STANDARD LIBRARY

Exercise Files

Starter – "engine/exercises/chapter10/exer_006.cc"
Answers – "engine/answers/chapter10/exer_006.cc"

Exercise Mission

n/a

Special Steps

Please remember, when you modify the engine and compile, you must copy the new executable over to your Kit/ directory before you can run it and see the changes in the Kit (as instructed below).

Synopsis

In this exercise, we will test your ability to convert code from ANSI-C standard library to Torque Standard Library calls.

Prerequisites

1. *ch1_001.pdf "Using The Kit"*

Exercises

1 Conversion

Goal: Convert the supplied code (using ANSI standard library calls) into code using the Torque Standard Library.

Starter Code: You are provided with one console function body to start this exercise.

```
ConsoleFunction(ch10_exer_006, const char *, 3, 3,
                "ch10_exer_006( string1 , string2 )")
{
   char *newString = Con::getReturnBuffer(256);
   const char *string1 = argv[1];
   const char *string2 = argv[2];

   //  ...
}
```

This console function does a variety of things, including all of the following.

1. Takes two strings as arguments and concatenates the together.

```
// Concatenate the strings
S32   len1 = strlen(string1);
S32   len2 = strlen(string2);

if( (len1 + len2) > 255 ) return NULL;

dMemcpy(newString, string1, len1);

dMemcpy(newString + len1 , string2, len2);

newString[len1+len2] = '\0';
```

2. Converts all characters in the new combined string into upper-case letters.

```
// Convert all characters to upper case
for(int i = 0; i<256;i++)
{
   newString[i] = toupper(newString[i]);
}
```

USING THE TORQUE STANDARD LIBRARY

3. Scans the string and converts individual letters into "leetspeak". (Yes, this is a very rudimentary conversion, but hey, it's an example!)

```
// do rudimentary leetspeak conversion
for(int i = 0; i<256;i++)
{
   switch( newString[i] ) {
   case 'O':
      newString[i] = '0';
      break;
   case 'D':
      newString[i] = '0';
      break;

   case 'I':
      newString[i] = '1';
      break;
   case 'T':
      newString[i] = '1';
      break;

   case 'E':
      newString[i] = '3';
      break;
   case 'B':
      newString[i] = '3';
      break;

   case 'Z':
      newString[i] = '2';
      break;
   case 'R':
      newString[i] = '2';
      break;

   case 'G':
      newString[i] = '6';
      break;

   case 'L':
      newString[i] = '1';
      break;
   }
}
```

USING THE TORQUE STANDARD LIBRARY

Steps:

1. Locate any and all ANSI-C functions that have reciprocal Torque Standard Library functions and replace them.
2. Re-compile and test your changes.

Output Goal:

After you successfully compile your code, you can start the kit, and open the console (~). Then, you can run the following command and you should receive the listed output.

```
==>echo( ch10_exer_006("convert this to ", "leetspeak") );
C0NV321 1H1S 10 1331SP3AK
```

Questions:

1. Were you able to convert all ANSI-C functions to Torque Standard Library?
2. Is there a better way to do the capitalization code? If so, why don't you try improving it?

Hints:

1. Don't forget to use appendix B.6.2.