Program Submission Instructions:

- You must submit your source code file plus a README file
- The source code file must be submitted in Webcourses from the assignment page
- All source code must be in exactly one file of type .c, .cpp, or .java
- The README file must be a text file

CIS 3360 – Security in Computing
Spring 2016
Program #2: CRC Codes (100 points)

Write a program that calculates the CRC-15 value for a given file and which can also verify the correctness of a given file that already has a CRC-15 value appended to it, as more fully described below.

Use the CRC polynomial: $x^{15}+x^{13}+ x^6+x^4+x^1+1$.

Program operation:

1. The program must compile from the command line.
2. The program executable file name must be "**crcheck**".
3. The program must run from the command line and take two (2) command line parameters.
4. The first command line parameter will be a flag value that identifies the mode of operation: "c" for calculating a CRC value, or "v" for verifying a CRC value. Only these two values are allowed. Any other values should produce a simple error message and a graceful exit from the program.
5. The second command line parameter will be the name of the file to be examined. The file should be a text file that is in the same folder as the program executable. If the file is not found, the program should issue a simple error message and exit gracefully.
6. The program should direct all output to the command window (terminal) screen. The details of what to output are described below.

What to submit:

Submit a **single** source code file written in C, C++, or Java. No other languages are permitted.

- The source code file must be a .c, .cpp, or .java file.
- Put all classes, functions, and methods in the one file.
- If programming in Java, do not place the source in your own package so our test scripts can run without changes.
- If you are coding in C or C++, you must use only the standard libraries, such as stdio.h, math.h, and Standard Template Library.
- If using Java, your source file must be named crcheck.java.

You must also submit a **README** file, which must be a text file, and which contains:

- The compilation command for your program
- The run command for your program
- ***Your statement that the program is entirely your own work and that you have neither developed your code together with any another person, nor copied program code from any other person, nor permitted your code to be copied or otherwise used by any other person, nor have you copied, modified, or otherwise used program code that you have found in any external source, including but not limited to, online sources.***

Input file format:

Valid input files will be ASCII files that contain printable data. There will be no pad or fill characters in a raw input file.

If an input file contains any invalid data (other than an end-of-file marker), then the program should issue an appropriate brief error message and terminate gracefully.

The raw input file will consist of ASCII data of varying length up to 512 bytes, with the last 8 bytes reserved for the checksum. IF the checksum consists of blank spaces, that is the last 8 bytes are **blank spaces,** then the CRC has NOT been calculated. (Note: The CRC or checksum will be a 16 bit integer, or 4 hexadecimal digits, which will fill the last 4 bytes/characters, the leading 4 characters should be zeroes.)

Output format:

Output the ASCII file that is read, 64 characters to a line. Alphabetic characters in this output must be represented as it was read from the input file, regardless of case.

Next, you will show the result of each 64 characters line's *cumulative* XOR operation involved in the CRC calculation or verification. For example:

```
abcdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345a - 0000206c
```

Note that the input is shown in black and the line's CRC is shown in **red**. (There is no need to use color in the output for the assignment.)

At the end, when calculating CRC, you must show the CRC result in hexadecimal. Please note that in the event the input file is less than 512 characters, you must pad the remainder of the output file with a period character (hex value 2E) reserving the last 8 characters for the ASCII representation of the CRC, with leading zeroes as needed to obtain a 16-bit value for the CRC-16 code.

On the other hand, if verifying CRC, you must output:

(a) the accumulated CRC value in hexadecimal at the end of each 64 character output of the input file including a pad of a period character (hex value 2E) and the stated hexadecimal CRC (Note that the CRC even though it is hexadecimal data it will be printed in ASCII.);

(b) the CRC (hexadecimal) calculated by the program; and

(c) a message whether the CRC check passed or failed.

Specific Functions inside code:

You must implement the following:

- A function/method to read the data in the input file into an array.
- A function/method for CRC calculation. This could be a function/method that takes as input the input data and returns the CRC result.
- A function/method for CRC verification, likewise,

Note:

- (Java does not support 16 bit unsigned integers unless using Java 8, hence the 32 bit unsigned integer appropriately used, provides adequate bit space for the CRC calculations.)

Grading Rubric

The total possible score for this program is 100 points. The following point values will be deducted for the reasons stated:

[-100 points]	**Your program does not successfully compile from the command line with one of these commands:** **C program: prompt> gcc –lm –o crcheck [your_file_name].c** **C++ program: prompt> g++ -lm –o crcheck [your_file_name].cpp** **Java program: prompt> javac crcheck.java** Note: If you are submitting a Java program, the class file must be named "crcheck.java" and the class name must be "crcheck" (all lower case).
[-90]	**Fails during execution**
[-50]	**Reads input file – no output**
[-25]	**Reads & displays input file – produce output but with invalid or null CRC (using the "c" mode).**
[-25]	**Reads & displays input file – does not detect invalid or null CRC (using the "V" mode).**
[-0]	**Reads input file, produces valid output, with correct CRC and the validation option also works correctly.**

Sample test runs

```
[536] mmcalpin@Hydrogen:~/UCF/CIS3360$java crcheck c WC-ngi.txt
CRC16 Input text from file

"This is the lesson: never give in, never give in, never, never,
 never, never - in nothing, great or small, large or petty - nev
er give in except to convictions of honour and good sense. Never
 yield to force; never yield to the apparently overwhelming migh
t of the enemy."

CRC 16 calculation progress:

"This is the lesson: never give in, never give in, never, never, - 000015fa
 never, never - in nothing, great or small, large or petty - nev - 00006206
er give in except to convictions of honour and good sense. Never - 00002dd2
 yield to force; never yield to the apparently overwhelming migh - 000007bf
t of the enemy."..... - 0000702e
..... - 000024a1
..... - 00005be7
.....00002db9 - 00002db9

CRC16 result : 00002db9
```

Figure 1.1 Program crcheck – Calculation option

```
[537] mmcalpin@Hydrogen:~/UCF/CIS3360$java crcheck v WC-ngi.crc
CRC16 Input text from file

"This is the lesson: never give in, never give in, never, never,
 never, never - in nothing, great or small, large or petty - nev
er give in except to convictions of honour and good sense. Never
 yield to force; never yield to the apparently overwhelming migh
t of the enemy."..........
..........
..........
.....00002db9


CRC 16 calculation progress:

"This is the lesson: never give in, never give in, never, never, - 000015fa
 never, never - in nothing, great or small, large or petty - nev - 00006206
er give in except to convictions of honour and good sense. Never - 00002dd2
 yield to force; never yield to the apparently overwhelming migh - 000007bf
t of the enemy."..... - 0000702e
..... - 000024a1
..... - 00005be7
.....00002db9 - 00002db9

CRC16 result : 00002db9

CRC 16 verification passed
```

Figure 1.2 Program crcheck – Verification option

Sample test runs

```
[535] mmcalpin@Hydrogen:~/UCF/CIS3360$java crcheck c input2A.plain
CRC16 Input text from file

abcdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345a
bcdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345ab
cdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abc
defghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abcd
efghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abcde
fghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345

CRC 16 calculation progress:

abcdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345a - 00001a6a
bcdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345ab - 00007d1d
cdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abc - 00001fdf
defghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abcd - 00000d90
efghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abcde - 0000259f
fghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345..... - 00003ef2
..................................................................... - 000057ad
.....................................................000075dc - 000075dc

CRC16 result : 000075dc
```

Figure 2.1 Program crcheck – Calculation option

```
mmcalpin@Hydrogen:~/UCF/CIS3360$java crcheck v input2A.crc
CRC16 Input text from file

abcdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345a
bcdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345ab
cdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abc
defghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abcd
efghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abcde
fghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345.....
.....................................................................
.....................................................000075dc


CRC 16 calculation progress:

abcdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345a - 00001a6a
bcdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345ab - 00007d1d
cdefghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abc - 00001fdf
defghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abcd - 00000d90
efghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345abcde - 0000259f
fghijklmnopqrstuvwxyz12345-ABCDEFGHIJKLMNOPQRSTUVWXYZ12345..... - 00003ef2
..................................................................... - 000057ad
.....................................................000075dc - 000075dc

CRC16 result : 000075dc

CRC 16 verification passed
```

Figure 2.2 Program crcheck – Verification option