Program Submission Instructions:

- You must submit your source code 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

CIS 3360 – Security in Computing
Spring 2018
HW #2 Checksum (100 points)

In this assignment you'll write a program that calculates the checksum for the text in a file. Your program will take two command line parameters. The first parameter will be for the size of the checksum (8, 16, or 32 bits). The second parameter will be the name of the input file for calculating the checksum. The program must generate output to the console (terminal) screen as specified below.

Command Line Parameters

1. Your program **must** compile and run from the command line.
2. The program executable must be named "checksum" (all lower case, no spaces or file extension).
3. Input the required file names as command line parameters. Your program may NOT prompt the user to enter the file names. The **first parameter** must be the size, in bits, of the checksum. The second **parameter** must be the name of the file used for calculating the checksum, as described below. The sample run command near the end of this document contains an example of how the parameters will be entered.
4. Your program should open the two files, echo the processed input to the screen, make the necessary calculations, and then output the ciphertext to the console (terminal) screen in the format described below.

Checksum size

The checksum size is a single integer, passed as the first command line argument. The valid values are the size of the checksum which can be either 8, 16, or 32 bits. Therefore, if the first parameter is not one of the valid values, the program should advise the user that the value is incorrect with a message formatted as shown below:

The number X is not a valid checksum size: valid checksum sizes are 8, 16, or 32.

The message should be sent to STDERR. (Note that the number **X** shown above should be the invalid number entered on the command line.)

Format of the input file

The input file will consist of the valid ASCII characters associated with the average text file. This includes punctuation, numbers, special characters, and whitespace.

Output Format

The program must output the following to the console (terminal) screen:

1. Echo the input file
2. Print the checksum.

The echoed input text should be in rows of exactly 80 letters per row, except for the last row, which may possibly have fewer. These characters should correspond to the input text. The checksum line should be formatted as follows:

The X bit checksum for the input file named xyz.ext is Z.

Where X is the checksum size of 8, 16, or 32 and the filename xyz.ext is the input filename and Z is the calculated checksum.

What to submit to WebCourses

You must submit this assignment's source code, appropriately commented, via WebCourses.

Program Notes and Hints

Your program must read in an input text file that may contain uppercase letters, lowercase letters and non-letter characters. Your program should then calculate the checksum appropriately for the size specified in the command line. Specifically, if the checksum is 8 bits long, each character should be used as the number to be added to the checksum. Likewise, if the checksum is 16 bits long, each two characters should be added to the checksum. Note that there is a 50% chance that there will be one character short on the input file. In that case use the character "X" (an uppercase X) as the pad character. Similarly, if the checksum is 32 bits, use the same technique and character to pad the input string appropriately.

Sample Run Command

C or C++ program:

```
      Prompt$ ./checksum 8 inputText1.txt
```

Java program:

```
      Prompt$ java checksum 8 inputText1.txt
```

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 –o checksum checksum.c
    C++ program:    prompt>     g++ –o checksum checksum.cpp
    Java program:   prompt>     javac checksum.java
```

Note: If you are submitting a Java program, the class file must be named “checksum.java” and the class name must be “checksum”.

[-100 point] The program does not accept input file names from the command line.

[-90 points] Your program does not run from the command line without error or produces no output.

[-70 points] The program compiles, runs, and outputs the input file (with a maximum of 80 characters per line), but crashes thereafter or produces no checksum output.

[-25 points] The program compiles, runs, echoes the input, and calculates the checksum, but the checksum output is incorrect and it is not formatted correctly (not all input text or not a maximum of 80 characters per line).

[no deductions] The program compiles, runs, echoes the inputs, generates encryption output, the encryption output is correct (ignoring case), and it is formatted correctly (all lowercase letters, 80 letters per line).