**What do you need help with?**
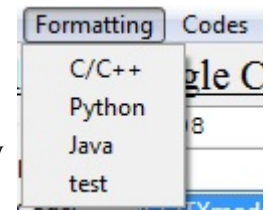
1. Drop Down Menu Options
2. Explanation of the menus and buttons
3. Calculating a single CRC Remainder
4. Bulk CRC Calculations
5. CRC Error Codes

1. Explanation of Menus and Buttons

1. Drop Down menu options

1. File Dropdown

   Save to folder does the same thing as the File Location browse button. Allows you to select the location where the Bulk CRC remainder table is saved The default location is in the output folder located in the program directory.
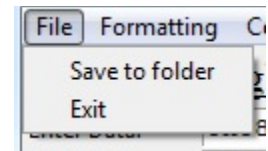

2. Formatting Dropdown

   Formatting dropdown allows you to prefill the output formatting entries for different commonly used programming languages. There is a text file in the program directory you can modify to add, modify or remove formatting options.

3. Codes Dropdown

   The Codes dropdown provides a list of commonly used CRC codes. You can select from this dropdown to pre-fill the CRC codes dropdown menues.
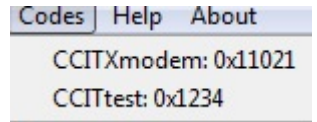
4. Help Dropdown

   This drowndown provides you the option of opening up an CRC explanation PDF, or this help file itself.
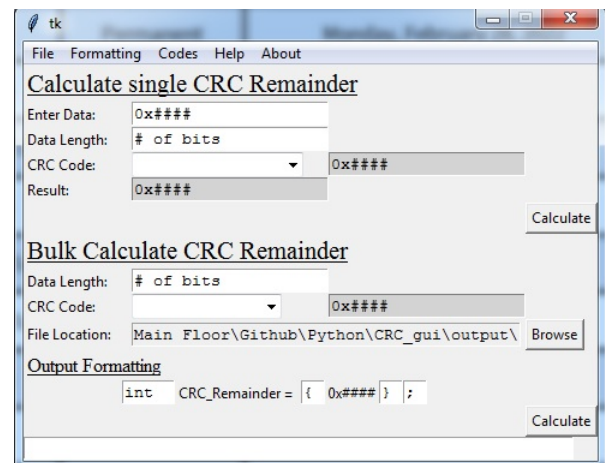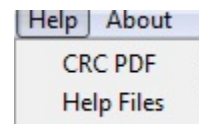
2. **Calculating single CRC Remainder**

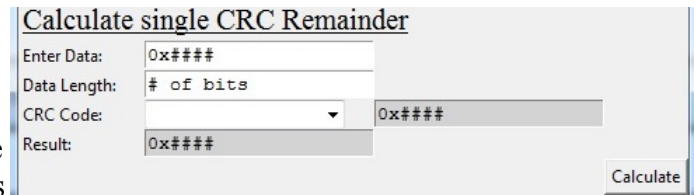Calculating a single CRC Remainder is to perform the CRC operations on a single chunk of data. You probably won't use this much, but you may want to calculate things out to test your system or do validation.

1. Enter Data: 0x####

    Requires a hexadecimal number. This GUI doesn't do the conversion of your data into hexadecimal format. However, there are a number of tools out there that can help you do this. Your data needs a standard hexadecimal prefix of "0x". "0x" is Zero than an x. Example:

Binary	Decimal	Hexadecimal
0000 0101	5	0x05
0000 1110	14	0x0D
1111 1111	255	0xFF

2. Data Length: # of bits

    Data length simply refers to the number of bits in your data. Your data could be 0x01. However, the length of the data chunk itself might be 4 bits (0001), 8 bits (0000 0001), or any other length imaginable. Therefor, to properly perform the CRC operation, the program needs to know how many binary digits are in one chunk of data.

3. CRC Code:

    The CRC code is important. Different codes will divide out and create different remainders. Furthermore, most CRC codes are recorded in a pre-truncated form. So, it is important to understand your CRC code. If in doubt, your CRC code is probably recorded in a pre-truncated form. To be sure, check its polynomial form.

Some common CRC's

CRC Name	Polynomial	Code
CRC – 8 – CCITT	$x^8 + x^2 + x + 1$	0x107
CRC – 12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$	0x80F
CRC – 16 – CCIT ("Xmodem")	$x^{12} + x^{11} + x^3 + x^2 + x + 1$	0x11021
CRC – 16 – IBM ("Modbus")	$x^{16} + x^{15} + x^2 + 1$	0x18005
CRC – 32 ("Zmodem")	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	0x104C11DB7

4. Result
   The result of a CRC operation is given in hexadecimal form.

3. Bulk Calculate CRC Remainder

Bulk Calculate CRC Remainder
Data Length: # of bits
CRC Code: ▼   0x####
File Location: Main Floor\Github\Python\CRC_gui\output\ Browse
Output Formatting
int CRC_Remainder = { 0x#### } ;
Calculate

Bulk CRC calculation doesn't look at any data chunk in specific. Rather, it runs through all possible data combinations for data of a particular binary length. Starting from 0, going up to the maximum number possible. The results are then formatted into a table and saved to a text file.

People often use table lookup instead of calculating the CRC remainder from scratch every time.

1. Data Length: # of bits

Data length simply refers to the number of bits in your data. You should know this in advance. Usually, your data will be some number of bytes long. For instance, if each data chunk you wish to send may be 1 byte or 8 bits long. It could also be 2 bytes or 16 bits long.

2. CRC Code

The CRC code is important. Different codes will divide out and create different remainders. Furthermore, most CRC codes are recorded in a pre-truncated form. So, it is important to understand your CRC code. If in doubt, your CRC code is probably recorded in a pre-truncated form. To be sure, check its polynomial form.
Some common CRC's

CRC Name	Polynomial	Code
CRC – 8 – CCITT	$x^8 + x^2 + x + 1$	0x107
CRC – 12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$	0x80F
CRC – 16 – CCIT ("Xmodem")	$x^{12} + x^{11} + x^3 + x^2 + x + 1$	0x11021
CRC – 16 – IBM ("Modbus")	$x^{16} + x^{15} + x^2 + 1$	0x18005
CRC – 32 ("Zmodem")	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	0x104C11DB7

3. File Location

File location refers to where on the hard drive the formatted output is saved. The default location is in an folder called "output" located in the same location as the CRC program itself. However, you can select any location if you so wish.

4. Output Formatting

The program automatically generates a table. This is useful for implementation in different programming languages. Tailor the output table to your needs using the output formatting entry boxes.

4. <u>Bottom Toolbar</u>

The bottom toolbar provides feedback information after you clock "calculate".

Code	Code Type	Code Description
Error	CRC Calculation Error	Something went wrong with the underlying CRC mathematics.
Warning	Input Error: Improper User Entry in "Enter Data" Data	This error code means that the input to the Enter Data text box was incorrect. Either the input was not a number, or the hexadecimal number formatting was incorrect.
	Input Error: Improper User Entry in "Data Length" Data	This error code means that the input into the Data length text box was incorrect. The input was not a base 10 integer number.
	Input Error: Improper User Entry in "CRC Code"	This error code means that the input to the CRC code text box was incorrect. The input was not a number, or the hexadecimal number formatting was incorrect.
	Input Error: Formatting option 1	This error means that the first text box, the variable type formatting, was incorrect. It would be hard to make this entry incorrect. Perhaps an unrecognized symbol would cause an error. Realistically anything you can type in with your keyboard should be allowable.
	Input Error: Formatting option "Pre Array syntax"	This error means that the second text box, the array bracket type formatting, was incorrect. It would be hard to make this entry incorrect. Perhaps an unrecognized symbol would cause an error. Realistically anything you can type in with your keyboard should be allowable.
	Input Error: Formatting Option "Post Array syntax"	This error means that the third text box, the array bracket type formatting, was incorrect. It would be hard to make this entry incorrect. Perhaps an unrecognized symbol would cause an error. Realistically anything you can type in with your keyboard should be allowable.
	Input Error: Formatting Option "After Array syntax"	This error means that the fourth text box, the array line end formatting, was incorrect. It would be hard to make this entry incorrect. Perhaps an unrecognized symbol would cause an error. Realistically anything you can type in with your keyboard should be allowable.
	Folder Error: Problem with "Output" folder	The output folder doesn't exist and or cannot be made in the specified location.
	Folder Error: Problem with "Log" folder	The error logging folder doesn't exist and or cannot be made in the specified location
	File Error: Problem with log file	The error logging file doesn't exist, cannot be made and or experienced an open or close error.

	File Error: Problem with output file	The output file doesn't exist, cannot be made and or experienced an open or close error.