

February 26, 2003

Coffee or Tea (10 points)

- 1. Create a project with a single form
- 2. On the form, place a command button. When the command button is clicked, invoke a common dialog box to change the background color of the form.
- 3. Also on the form, create an additional command button which, when clicked, displays an Input Box that will ask the user if they would prefer coffee or tea. Set a default value of "Coffee", and position the Input Box in the upper left hand corner of the screen.
- 4. If the user responds correctly with either coffee or tea, a message should be displayed informing the user that their coffee (or tea, whichever they chose) will be ready in five minutes. If the user response is incorrect, the input box should be redisplayed. This process should continue until a correct response is obtained.



The Newport News Computer Challenge February 26, 2003

Roman Numeral Calculator (30 points) Part 1, Design Phase (10 of the 30 points) Part 2, Logic Phase (20 of the 30 points)

Folder Name:	RomanCalc	
Files to be place	ed in folder:	
Form H	File Name:	RomanCalc.frm
Project	t File Name:	RomanCalc.vbp

Design a Roman Numeral Calculator that looks similar to this one:

Roman Numeral Calculator			⊐ ×	
Roman Nur	/eral C	ALCULATO	2	
MCCXXVI				
I X C Clear	V L D	/ * - +		

When the program starts, the text box should be empty.

In **Part 1, the Design Phase**, buttons labeled **I**, **V**, **X**, **L**, **C**, **D**, **M**, **and Clear** should work properly so that the user can click them and thereby enter a Roman Numeral into the text box. Entry validation is not required to get full credit for Part 1.

In **Part 2, the Logic Phase**, buttons labeled /, *, -, +, and = should work properly. Since there is no backspace key, once a Roman Numeral digit is entered into the text box, the text box should never clear except when the Clear key is pressed or a digit is entered immediately after the = key is pressed.

Division is to be computed as integer division.

Roman Numerals cannot represent zero or negative integers. Therefore, if the result of an operation would result in zero or a negative integer, the text box should display "Error". Once "Error" is displayed, no keys should work except the Clear key.

No message boxes should be used in the program.

The final 5 points will be awarded to a program that works correctly (all buttons work correctly) AND invalid Roman Numerals are prevented from being entered by a digit key (I, V, X, L, C, D, M) not responding. For example, if the user presses X (displaying an X in the text box) and then the user pressed M, the M key will not respond because XM is an invalid Roman Numeral.

The values of Roman digits are as follows:

 $\begin{array}{cccc} I & 1 \\ V & 5 \\ X & 10 \\ L & 50 \\ C & 100 \\ D & 500 \end{array}$

M 1000

Here are the rules for forming Roman Numerals:

- 1) Digits are combined left to right, larger to smaller.
- 2) Digits for powers of 10 (I, X, C, and M) may be written 1, 2, or 3 times in succession. Other digits may appear only once in a Roman Numeral.
- 3) Numerals are evaluated by adding the values of their digits.
- 4) Certain exceptions allow a smaller to precede a larger, which means subtract the smaller from the larger:
 - a) I may precede V which means 4 (5 1).
 - b) I may precede X which means 9 (10 1).
 - c) X may precede L which means 40 (50 10).
 - d) X may precede C which means 90 (100 10).
 - e) C may precede D which means 400 (500 100).
 - f) C may precede M which means 900 (1000 100).
- 5) Thus, the Roman Numerals for 1 to 9 are I, II, III, IV, V, VI, VII, VIII, and IX.
 - a) Roman Numerals for 10 to 90 (counting by 10) are the same except use X, L, and C.
 - b) Roman Numerals for 100 to 900 (counting by 100) are the same except use C, D, and M.
 - c) Roman Numerals for 1000 to 3000 (counting by 1000) are M, MM, and MMM.
- 6) The largest Roman Numeral using this system is MMMCMXCIX, which equals 3999.
- 7) There are no Roman Numerals for zero or negative integers.

Notice that rules 5, 6, and 7 do not provide any new information. They simply clarify rules 1, 2, 3, and 4.

One variation of the rules for forming Roman Numerals is to allow 4 I's, X's, C's, or M's in succession, thus eliminating the need for the exceptions that require subtraction (IV, IX, etc.). Sorry, but this variation is not allowed in this problem.