

Test Plan for Black Box tests of Roman Numeral Program

**26th oct 2019**

**Written by:**

**CS416: Section 3 – Group 7**

**Tony Moraes**

**Jacob Thomas**

**Nick Campbell**

**Ryan Dockstader**

# Scope

This job intent to provide a test plan for the Roman numeral program using the Black Box Testing technique. That will cover the aspect of the software that could be used by the final user and could be reproductible. This document does not cover the results as a test report would do.

The requirements that will be tested was listed from the following definition:

* Only two forms of input: a Roman numeral or a command to quit
* There is only one prompt used both for the Roman numeral and for the quit command
* Case is ignored for all input. Thus, "mmm" is the same as "MMM"
* Do not output anything if the quit command is input
* Output "valid" if the input is a valid Roman numeral. Otherwise, output "invalid"

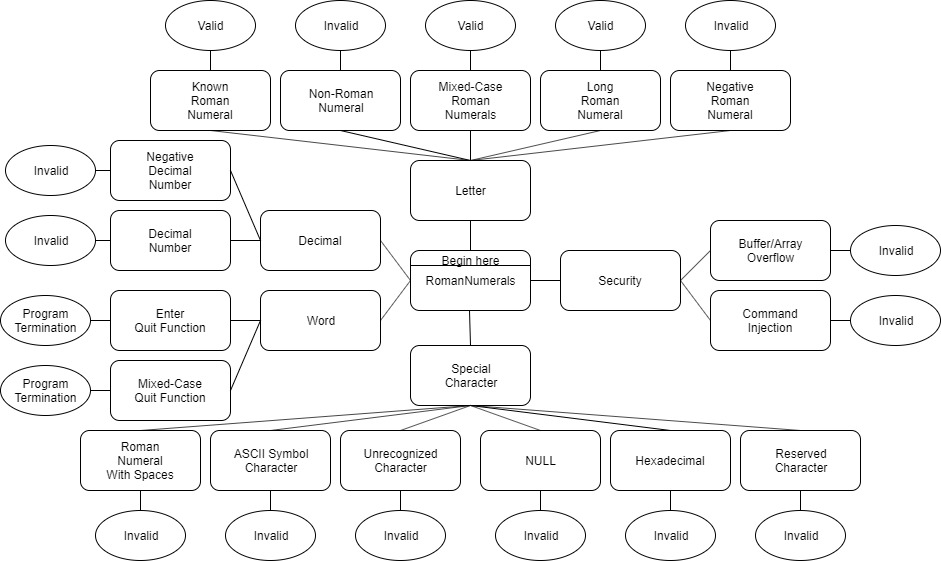
# Approach

Following the requirements specifications, a test graph was built the test cases built from them. Those test cases should be performed manually.

# Resources

For such small program a unique person could perform the tests provided in this document. Also, a definitive list of valid Roman numerals for our program is available on the Linux system at the following location: /home/cs416/week06/RomanNumerals.txt

## Test Graph



## Test Cases

### Letter

|  |  |
| --- | --- |
| **Title: Check actual roman numeral letters** | |
| **Steps:** | Enter a roman numeral letter |
| **Expected output:** | “Valid" appears |
| **Title: Check against generic letters** | |
| **Steps** | Enter a non-roman numeral letter |
| **Expected output** | "Invalid" appears |
| **Title: Check against case sensitive letters** | |
| **Steps** | Enter uppercase roman numeral letter |
| **Expected output** | “Valid" appears |
| **Title: Check against length of valid roman numeral** | |
| **Steps** | Enter very long roman numeral letter (MMMCMXCIX) |
| **Expected output** | “Valid” appears |
| **Title: Check negative input** | |
| **Steps** | Enter a negative roman numeral letter (-L) |
| **Expected output** | “Invalid” appears |

### Decimal

|  |  |
| --- | --- |
| **Title: Check against numbers** | |
| **Steps** | Enter a decimal number |
| **Expected output** | "Invalid" appears |
| **Title: Check against Negative decimal** | |
| **Steps** | Enter -1 |
| **Expected output** | “Invalid” appears |

### Special Character

|  |  |
| --- | --- |
| **Title: Check against gaps in input** | |
| **Steps** | Enter space between two numerals |
| **Expected output** | “Invalid” appears |
| **Title: Check against ASCII characters** | |
| **Steps** | Input the ASCII code for any character |
| **Expected output** | “Invalid” appears |
| **Title: Check against unrecognized characters** | |
| **Steps** | Input an unrecognized character (such as a Chinese character without the Chinese language installed on the test system) |
| **Expected output** | “Invalid” appears |
| **Title: Check against NULL input** | |
| **Steps** | Input nothing and submit |
| **Expected output** | “Invalid” appears |
| **Title: Check against Hexadecimal input** | |
| **Steps** | Input a hex code |
| **Expected output** | “Invalid” appears |
| **Title: Check against Reserved character input** | |
| **Steps** | Check against a reserved character, such as \n |
| **Expected output** | “Invalid” appears |

### Word

|  |  |
| --- | --- |
| **Title: Check the quit function** | |
| **Steps** | Enter 'quit' |
| **Expected output** | Program termination |
| **Title: Check against case sensitive commands** | |
| **Steps** | Enter 'QUIT', 'Quit', or ‘QuIt’ |
| **Expected output** | Program termination |

### Security

|  |  |
| --- | --- |
| **Title: Check against long strings (buffer/array overflow)** | |
| **Steps** | Enter a very long string (1000+ characters) |
| **Expected output** | “Invalid” appears |
| **Title: Check against command injection** | |
| **Steps** | Initiate a command injection attack  (such as: ”‘NNMN’) command(…) “) |
| **Expected output** | “Invalid” appears |