Kyle Vecere

The following shows all the potential ways to erroneously input information into the program.

**Erroneous Input Tests (Row Input)**

**Type: Numbers and Spaces**

Input for row 0: 1234567 8

How it’s handled: Due to the use of cin, boardMaker reads the input up until the first whitespace. Will return an error unless the whitespace comes after the 9 non whitespace characters that you’ve entered.

Output to console: Error: Row 0 has an incorrect number of digits

**Type: Numbers and Various other Characters**

Input for row 0: 12345678@

How it’s handled: The input is read on a row by row, character by character basis. The program checks these characters’ ASCII value in boardMaker. boardMaker asks for input of same row again.

Output to console: Error: Illegal character in row 0.

**Type: Anything With a Length Other Than 9.**

Input for row 0: 1234567

How it’s handled: Every time a row is entered, boardMaker checks the number of characters extracted by the last stream read to see if it’s equal to 10 (9 numbers plus the carriage return). boardMaker asks for input of same row again.

Output to console: Error: Row 0 has an incorrect number of digits.

**Type: Conflicting Rows/Columns/Grids**

Input for row 0: 123456789

Input for row 1: 123456789

Input for row 2: 123456789

Input for rows 3-8: 000000000

How it’s handled: The program checks for duplicates in rows/columns/grids in getPossibleCombos, whereby if there is a duplicate, getPossibleCombos returns a c-string that contains a “!” as the first element in the array. Error is thrown in main function, and board making process starts over.

Output to console: Error: Duplicate in row 0, column 1.

You must enter your board information correctly.

**Erroneous Input Tests (Filename Input)**

Note: Referred to Microsoft’s Documentation regarding Filenames and Paths

Link: <https://msdn.microsoft.com/en-us/library/aa365247>

**Type: Anything With a Space or Period At the End**

Input for filename: hello.

How it’s handled: setFilenameValid checks the second to last character in the c-string to see if it’s a space or a period.

Output to console: Error: space/period at end of filename.

**Type: Reserved Filenames**

Input for filename: CON

How it’s handled: setFilenameValid compares the input with a c-string of filenames to see if they’re both the same length and have the same characters.

Output to console: Error: Filename is reserved

**Type: Filenames with Illegal Characters (ie @, !, \, etc.)**

Input for filename: hel\o

How it’s handled: setFilenameValid iterates through and compares the input c-string to a c-string of unacceptable characters.

Output to console: Error: Filename contains illegal character \

**Type Filenames Longer Than 256 Characters**

Input for filename: Lorem ipsum dolor sit amet, ut eum doming consequuntur. Ei case assum vel, aliquando assueverit scripserit eos an, te enim labitur qui. His dicam iudico imperdiet ad, duo eu ponderum elaboraret. At duo malorum alterum, falli tincidunt ut his

How it’s handled: setFilenameValid checks the length of the number of characters extracted by the last stream read to see if it’s greater than 257 (256 plus the carriage return)

Output to console: Error: Filename is too long.

**Erroneous Input Tests (Decision to Make Multiple Boards)**

**Type: Anything With a Length Greater Than 1**

Input for response: yes

How it’s handled: due to the use of cin (and the fact that it’s being stored as a char), only the first non-whitespace character is read.

Output: Enter info for row 0

**Type: Anything that doesn’t start with y or n**

Input for response: hello

How it’s handled: due to the use of cin (and the fact that it’s being stored as a char), only the first non-whitespace character is read. The main function checks the ascii value of the first character to see if it matches that of y or n.

Output: Invalid response. Please try again.