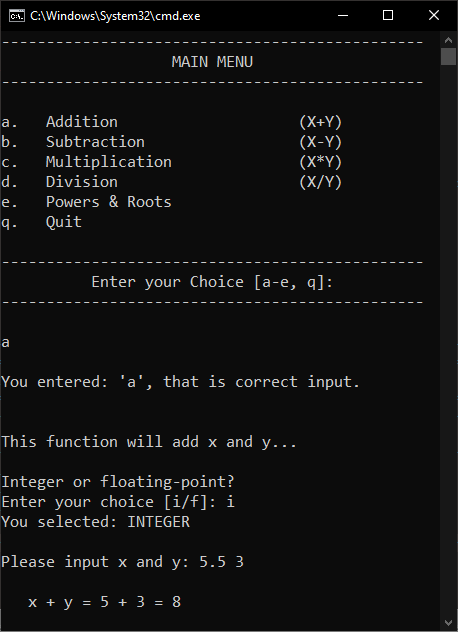
Michael Ring

Project 1 Screenshots

**1 – Welcome and Integer Addition**

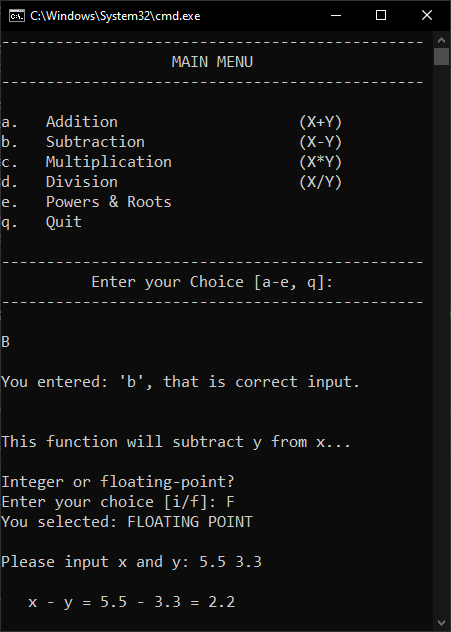
This is what the user sees when opening the program.

Inputting ‘a’ directs the program to the addition function.

The user is prompted to select int/float.

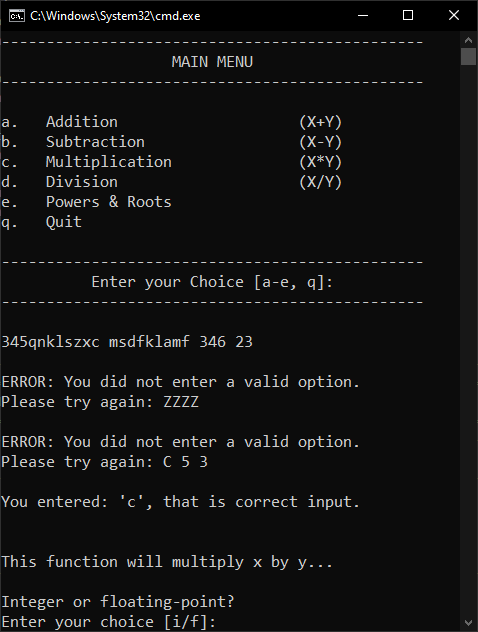
Inputting ‘i’ flags the addition function to convert the data type.

5.5 is truncated to 5.

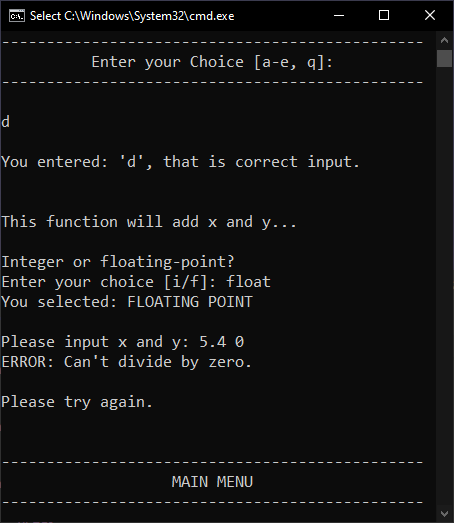
**2 – tolower() and Floating Point Subtraction**

Using uppercase letters is still recognized by the program.

Additionally, selecting floating point allows decimals to not be truncated.

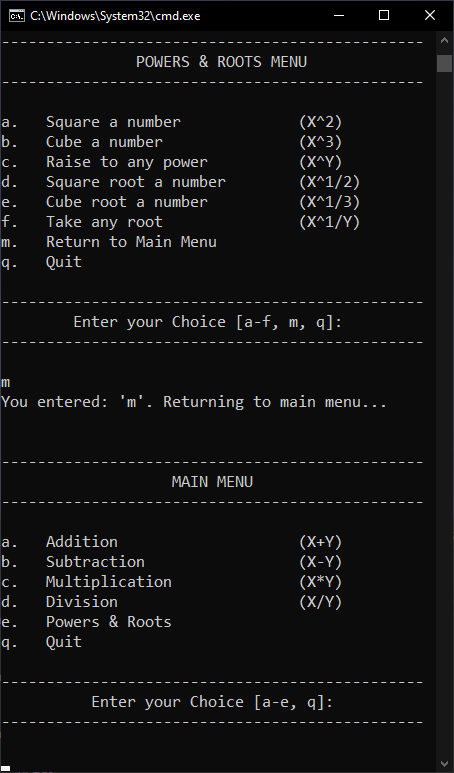
**3 – Main Menu Input Validation**

The main menu function rejects jumbled characters with spaces, number selections, invalid char selections, and ignores input that follows a valid selection.

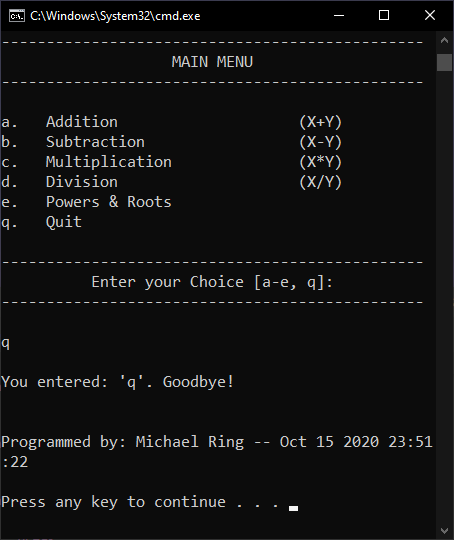
**4 – Divide by Zero Check**

Entering ‘float’ for i/f choice still recognizes input as ‘f’.

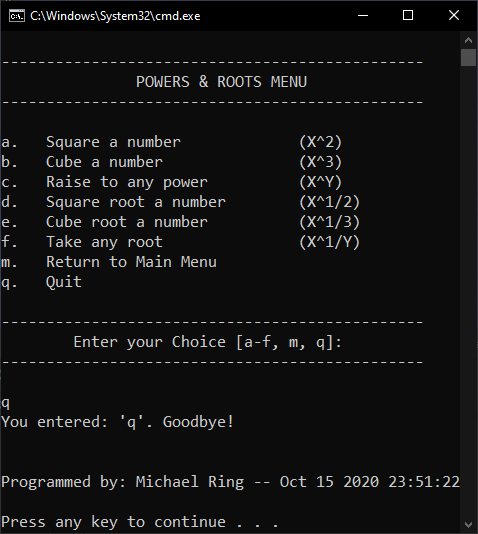
Trying to divide by zero results in an error message and redirects the user to the main menu.

**5 – Returning to Main Menu from Powers & Roots**

Entering ‘m’ closes the powers & roots loop, returning to the main menu (without quitting).

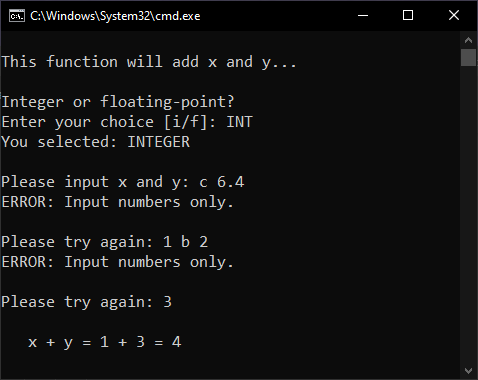
**6 – Quit from Main Menu**

Entering ‘q’ from the main menu prints the compile time, programmer name, date/time, then pauses.

**7 – Quitting from Powers/Roots Menu**

Entering ‘q’ from the Powers & Roots menu flags the main driver to end its loop, ending the program.

**8 – x and y input validation**

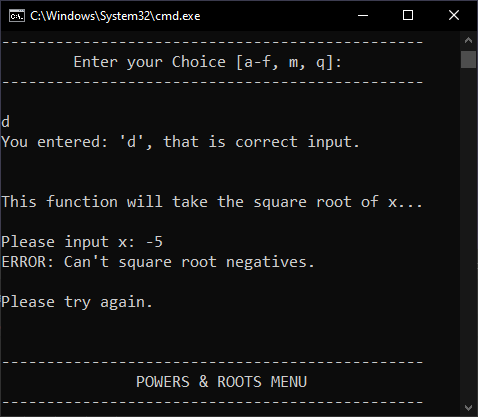
****

getXandY doesn’t read any input after characters.

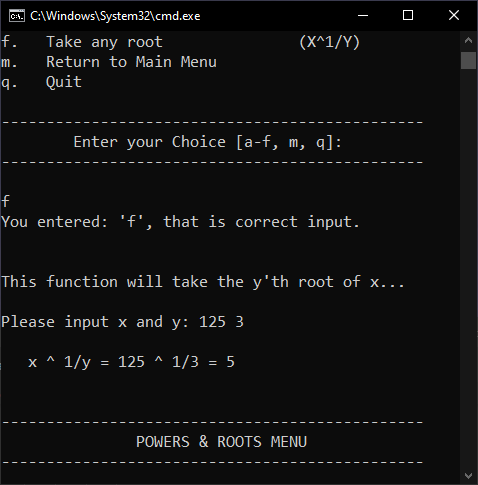
‘c 6.24’ doesn’t read anything

‘1 b 2’ reads only 1

‘3’ reads 3 for y

**9 – Square root of a negative**

Similarly to dividing by zero, inputting a negative number in the sqrt function throws an error before returning to the powers & roots menu.

**10 – Take any Root**

Functions as expected, and returns to pr menu afterwards.