==========================

=== UVSim Instructions ===

==========================

Confirm that Python 3 or later is installed.

Confirm that Pyqt6 is installed.

Run main.py to start the application

Select the File button, and select the test file wanted from its destination. The text file should

contain only properly formatted operation codes, six-digit signed integers, or four-digit signed integers such as +1234 and -5678 that are converted to six-digit signed integers.

The contents of the file selected will then display in memory

Select the Run button, this will trigger the function within the user's code editor terminal outside of the application

Input words as prompted to code editor terminal

Result is printed to the 'Console Output' widget in the application

The first three digits of the operation code specifies the operation selection.

The final three digits of the operation codes specify the memory location.

=================================

=== UVSim: BasicML Vocabulary ===

=================================

I/O operation:

READ = 010 Read a word from the keyboard into a specific location in memory.

WRITE = 011 Write a word from a specific location in memory to screen.

Load/store operations:

LOAD = 020 Load a word from a specific location in memory into the accumulator.

STORE = 021 Store a word from the accumulator into a specific location in memory.

Arithmetic operation:

ADD = 030 Add a word from a specific location in memory to the word in the accumulator (leave the result in the accumulator)

SUBTRACT = 031 Subtract a word from a specific location in memory from the word in the accumulator (leave the result in the accumulator)

DIVIDE = 032 Divide the word in the accumulator by a word from a specific location in memory (leave the result in the accumulator).

MULTIPLY = 033 multiply a word from a specific location in memory to the word in the accumulator (leave the result in the accumulator).

Control operation:

BRANCH = 040 Branch to a specific location in memory

BRANCHNEG = 041 Branch to a specific location in memory if the accumulator is negative.

BRANCHZERO = 042 Branch to a specific location in memory if the accumulator is zero.

HALT = 043 Pause the program