Running:
The program is run by invoking an installation of Python3, followed by Monitor.py, followed by an optional intex HEX filename.
EX:

```
Python3 main.py {test1.obj}
```

Functions:

Monitor.main : Gets user input, translates to values actually usable to call functions, then calls those functions on the emulator.

Emulator.parse_intel_hex : loads a given file, interprets it as an intex hex file, and then passes the parsed values to the memory setting function

Emulator.display_registers : prints the contents of the special registers in the format specified by the project writeup.

Emulator.initalize_registers : resets all special registers to 0

Emulator.save_values_to_memory : takes in a starting location and a list of memory values, then stores those values into memory starting at the starting location

Emulator.display_values_from_range : display the data in memory from the eight floored starting value to the ending value. Format with 8 values per line and the memory location as a tag before the eight values.

Emulator.display_single_values : display the data within memory at the given location

Emulator.get_hex_string_from_decimal_number: convert a base 10 value to an appropriately formatted hex string

Emulator.get_decimal_number_from_hex_string : get the decimal values of a string interpreted as hexadecimal

Emulator.prompt_for_input : get a normal value from user and return true, or get an exit code and return false

Testing:
Tests are located in test_functions.py. Static methods and conversions were tested to ensure no data was long going back and forth.
A selected batch of example runs of the program provided by the documentation were directly translated to tests. Command line object loading was intended to be added, but the intended

implementation would be operating system specific, and therefore testing accuracy would be ambiguous. Manual testing of all example test cases was performed before submission.