DOCUMENTATION

Procedure to Run:

In command prompt navigate to the directory where the code file is stored. Run the .py file on the terminal. User will be asked to enter the name of the input file once provided the correct input file, the output i.e. the Opcode Table, Machine Code, Symbol Table will be generated in the same directory as that of the code file.

Errors will be shown on the terminal.

Machine code will only be generated when there'll be no error.

Assumptions:

- There must be some instruction present after the label name.
  (For example - L1: DSP A)
- Address is assigned through location counter to avoid the problem of getting same address for two different variables/labels when

a large number of variables/labels are declared.
- Instructions can Start at any address.
- Since the address will be of 8-bit if location counter becomes greater than 256 it will throw “Overflow Error”.

First Pass:

In the first pass, the input file i.e. the assembly code is interpreted and then, a corresponding symbol table is generated through the assembler. When declarative statements are read, the variable of the opcode is directly allotted an address (8-bit random address). Branch statements are used to identify the labels, and the labels are put in the symbol table using forward referencing.

Second Pass:

The instruction opcodes and symbol table are used to read the assembly

code again and the corresponding machine code is generated.

Error Handling:

- Variable already declared.
- Overflow
- Limited number of opcodes provided.
- End of program not found
- Variables cannot be used as Labels.
- Label defined multiple times.
- Invalid Syntax.
- Empty Input File.
- Variable not declared.
- Labels declared cannot be used as Variables.

Output:

The following output files will be generated:

1. MACHINE CODE.txt
2. SYMBOL TABLE.txt
3. OPCODE TABLE.txt