**Documentation**

Opcodes:-

CLA/ 0000: Clear the accumulator,

LAC/ 0001: load content into accumulator,

SAC/ 0010: Store the contents of the accumulator ,

ADD/ 0011: Add register or memory to the accumulator,

SUB/ 0100: Subtract the register or the memory from the accumulator,

BRZ/ 0101: Branch if zero,

BRN/ 0110: Branch if negative,

BRP/ 0111: Branch if zero or positive,

INP/1000: Input character to accumulator,

DSP/ 1001: Display the content,

MUL/ 1010: Multiply register or memory to the accumulator,

DIV/1011: Divide accumulator content by the address content ,

STP/ 1100: Stop the execution

First Pass:

\\Assumption and working

Making an assumption that comments start with the hash character“#”

Assuming Symbol table only consists of labels, variable table consists of variables,Literal table consists of literals,opcode table consists of the order of opcode written in the code

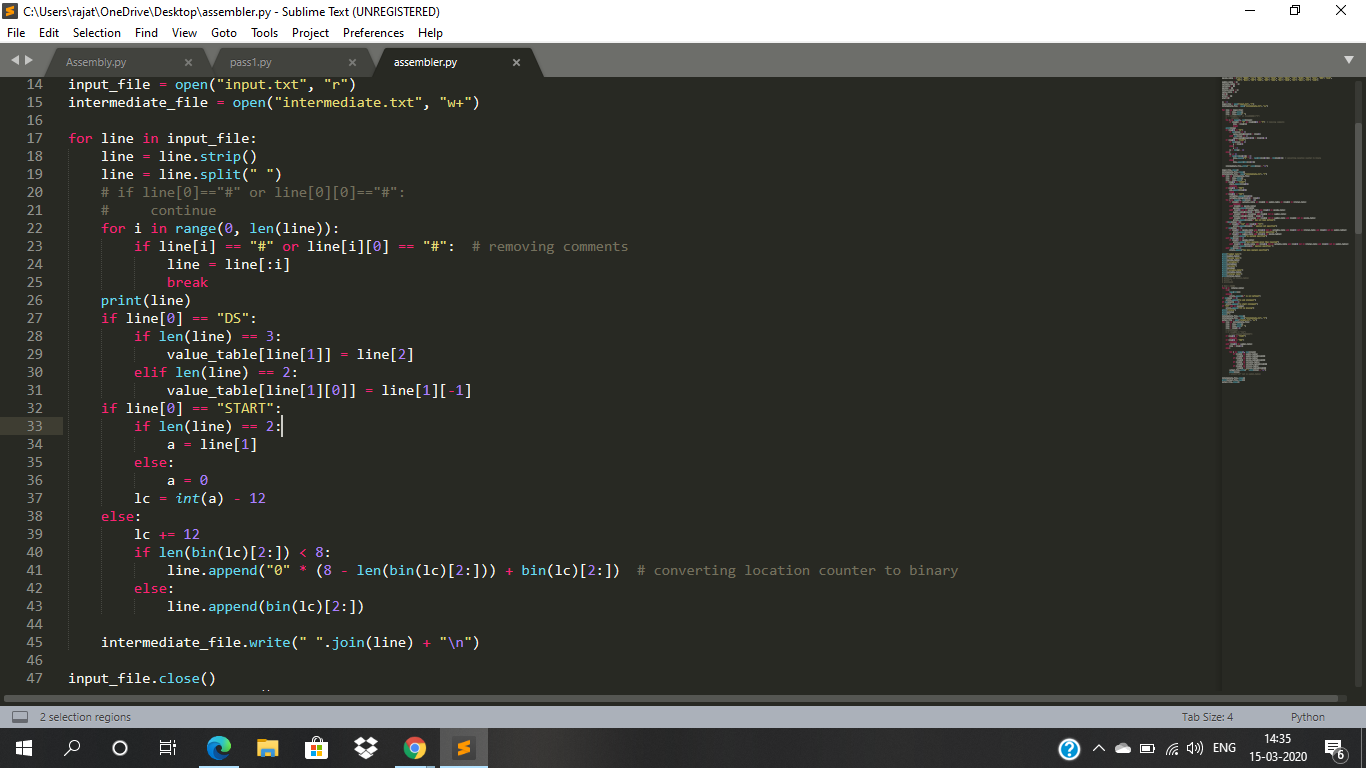
Assuming no variable or label has same name as the opcode define in assembler

Assuming label name and variable name is not same

First pass perform 3 main task:-

* Find the opcode, literal, variable and label from START to END .
* Build a Symbol table, opcode table, literal table and variable table for the following.
* Assign address to them by location counter

And then Intermediate code is created which is used by Second pass to create machine code



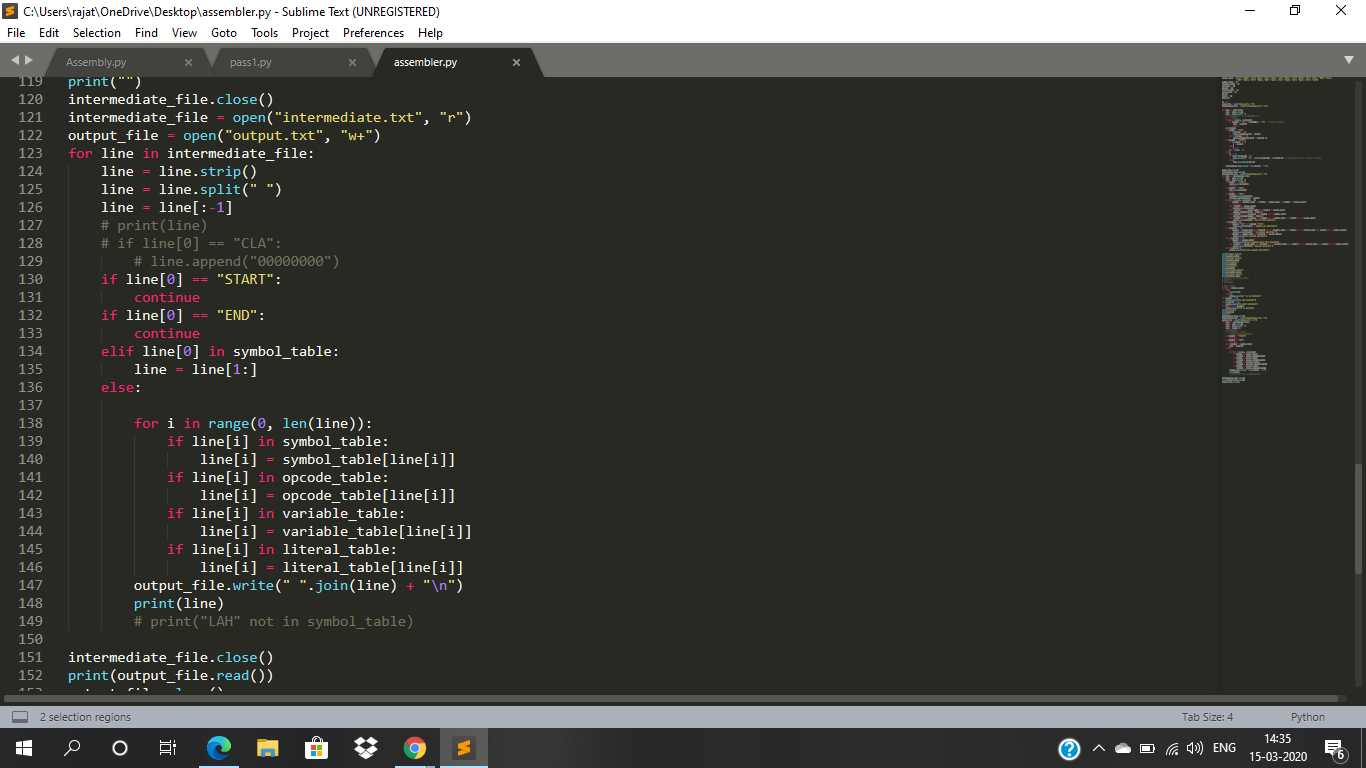
(Here is the Screenshot of the First pass code)

Second Pass:

\\working

Second pass takes Intermediate code as an input and line by line convert it into machine code by:-

* Converting the assembly opcode into their 4 bit binary representation
* Replacing literal, variable and label by their address using their corresponding tables

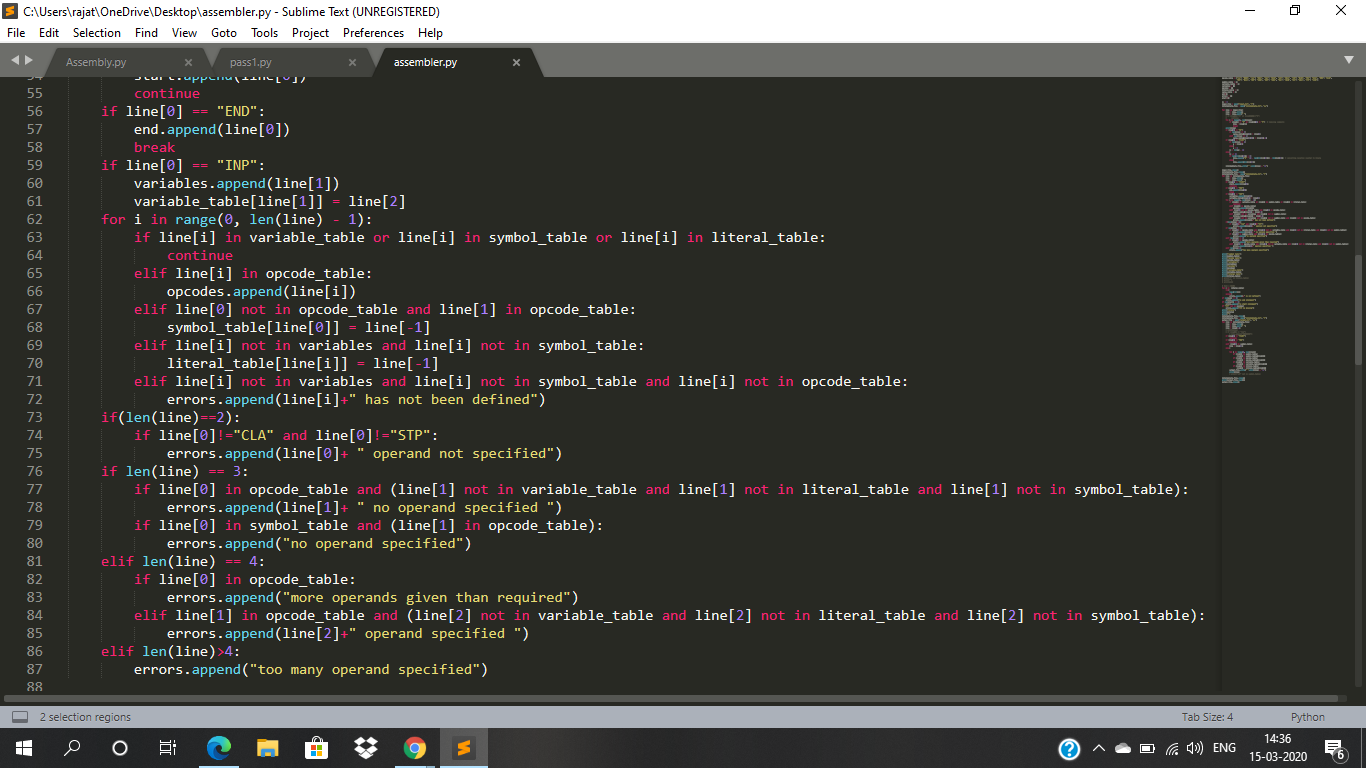


(Here is the Screenshot of Second pass)

Error Reporting:

\\explanation of errors

* If there is no “START” in the code then it will show an error
* If there is no “END” in the code then it will show an error
* If literal is not a integer type then it will show an error
* If “STP” opcode is not in the code then it will show an error
* If there is no operand is declared after opcode then it will an error
* Error will be shown if more than one operand is declared after opcode



(Here is the screenshot of Error reporting)