SS LAB 03

SAHIL BONDRE: U18CO021

Implement First Pass Assembler(Symbol Table,Literal Table,Pool Table and Table of Incomplete Instructions) for multiplication of two numbers

file.asm

```
  START 200
  READ X
  MOVER AREG, X
  MOVER BREG, ="5"
  MULT AREG, BREG
  MOVEM AREG, Y
  PRINT Y
  STOP
  LTORG
    ="5"
X DS 1
Y DS 1
  END
```

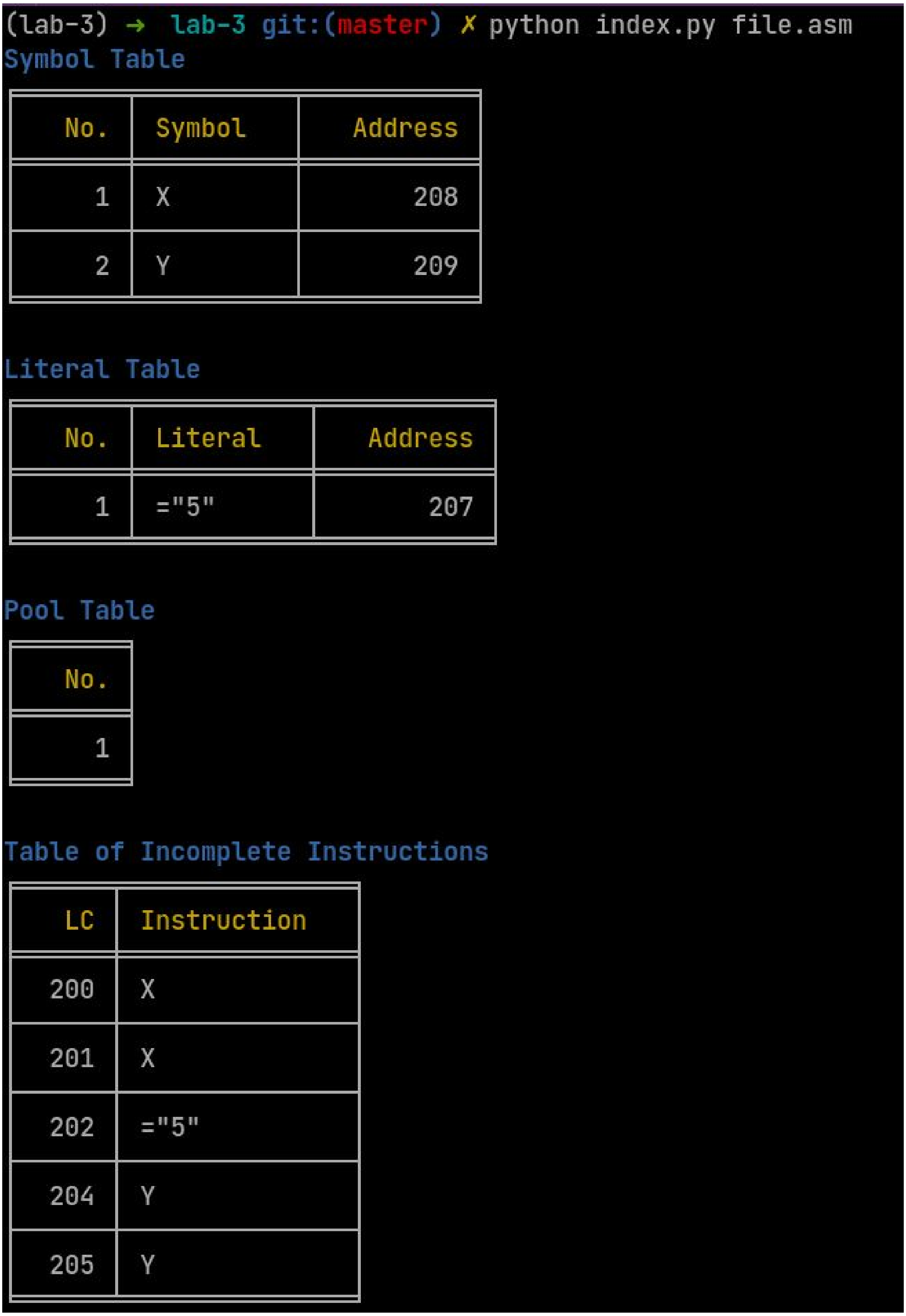
requirements.txt

```
Package    Version
---------- -------
pip        21.0.1
setuptools 53.0.0
tabulate   0.8.7
termcolor  1.1.0
wheel      0.36.2
```

index.py

```
import sys
import re
```

```
from termcolor import colored
from tabulate import tabulate

if len(sys.argv) != 2:
    print("Usage: python index.py <file-name>")
    exit(1)

file_name = sys.argv[1]

symbol_table = []  # no, symbol, address
literal_table = []  # no, literal, address
pool_table = []  # no
tii = []  # lc_no, instruction
incomplete_instructions = set()

address = 0
with open(file_name) as f:
    for num, line in enumerate(f, 1):
        is_label = not bool(re.match(r'\s', line))
        tokens = line.split()
        # print(tokens)
        # print(num)
        # print(is_label)
        op: str = tokens[0]
        if num == 1:
            address = int(tokens[1]) - 1
        elif is_label:
            symbol_table.append((len(symbol_table) + 1, op, address))
            incomplete_instructions.add(op)
        elif op.startswith("="):
            literal_table.append((len(literal_table) + 1, op, address))
            incomplete_instructions.add(op)
        elif len(tokens) >= 2 and not tokens[-1].endswith("REG") and
tokens[-1] not in incomplete_instructions:
            tii.append((address, tokens[-1]))
        elif op == "LTORG":
            address -= 1

        address += 1


def generate_pool_table(literal_table):
    pool_table = []
    for i, r in enumerate(literal_table):
        if i == 0:
            pool_table.append([r[0]])
        elif r[2] != 1 + literal_table[i - 1][2]:
            pool_table.append([r[0]])
    return pool_table


pool_table = generate_pool_table(literal_table)

print(colored("Symbol Table", attrs=["bold"], color="blue"))
print(tabulate(symbol_table, headers=[
      colored("No.", color="yellow"), colored("Symbol", color="yellow"),
colored("Address", color="yellow")], tablefmt="fancy_grid"))
print()

print(colored("Literal Table", attrs=["bold"], color="blue"))
print(tabulate(literal_table, headers=[
      colored("No.", color="yellow"), colored("Literal",
color="yellow"), colored("Address", color="yellow")],
tablefmt="fancy_grid"))
print()

print(colored("Pool Table", attrs=["bold"], color="blue"))
print(tabulate(pool_table, headers=[
      colored("No.", color="yellow")], tablefmt="fancy_grid"))
print()

print(colored("Table of Incomplete Instructions",
              attrs=["bold"], color="blue"))
print(tabulate(tii, headers=[
      colored("LC",  color="yellow"), colored("Instruction",
color="yellow")], tablefmt="fancy_grid"))
```



```
(lab-3) ➜  lab-3 git:(master) ✗ python index.py file.asm
Symbol Table
╒═══════╤══════════╤═══════════╕
│   No. │ Symbol   │   Address │
╞═══════╪══════════╪═══════════╡
│     1 │ X        │       208 │
├───────┼──────────┼───────────┤
│     2 │ Y        │       209 │
╘═══════╧══════════╧═══════════╛

Literal Table
╒═══════╤═══════════╤═══════════╕
│   No. │ Literal   │   Address │
╞═══════╪═══════════╪═══════════╡
│     1 │ ="5"      │       207 │
╘═══════╧═══════════╧═══════════╛

Pool Table
╒═══════╕
│   No. │
╞═══════╡
│     1 │
╘═══════╛

Table of Incomplete Instructions
╒══════╤═══════════════╕
│   LC │ Instruction   │
╞══════╪═══════════════╡
│  200 │ X             │
├──────┼───────────────┤
│  201 │ X             │
├──────┼───────────────┤
│  202 │ ="5"          │
├──────┼───────────────┤
│  204 │ Y             │
├──────┼───────────────┤
│  205 │ Y             │
╘══════╧═══════════════╛
```