# SS LAB 2

## SAHIL BONDRE: U18CO021

**Write a dynamic program to generate a Symbol Table with functions to create,insert, modify, search and display.**

**index.py**

```
import sys
import re
from termcolor import colored

from symbol_table import SymbolTable

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

file_name = sys.argv[1]


address = int(input("Enter starting address: "))

st = SymbolTable(address)
print("Symbol: Address")
with open(file_name) as f:
    lines = f.readlines()
    for line in lines:
        st.insert(line)

while True:
    c = input("""
Options:
  i: insert
  m: modify
  s: search
  d: display
  q: quit
  Your Choice: """)

    if c == "i":
        st.insert(input("Enter next instruction: "))
        print(colored(f"Inserted!", color="green"))
```

```
    elif c == "m":
        symbol = input("Enter symbol to modify: ")
        if symbol in st.symbols:
            addr = int(input("Enter new Address: "))
            for field in st.fields:
                if field.symbol == symbol:
                    field.address = addr
                    break
            print(colored(f"Modified!", color="green"))
        else:
            print(colored("Error: Symbol not found", color="red"))
    elif c == "s":
        symbol = input("Enter symbol to search: ")
        if symbol in st.symbols:
            for field in st.fields:
                if field.symbol == symbol:
                    print(colored(f"{symbol}: {field.address}",
color="green"))
                    break
        else:
            print(colored("Error: Symbol not found", color="red"))
    elif c == "d":
        st.print()
    elif c == "q":
        break
    else:
        print("Error: Unknown Command")
```

**symbol_table.py:**

```
import re
from symbol_field import SymbolField
from tabulate import tabulate
from termcolor import colored


class SymbolTable:
    def __init__(self, start: int):
        self.fields = []
        self.symbols = set()
        self.address = start

    def insert(self, line: str):
```

```
        if not re.match(r'\s', line):
            # starts with symbol
            symbol = line.split()[0]
            if symbol not in self.symbols:
                self.fields.append(SymbolField(symbol, self.address))
                self.symbols.add(symbol)
            else:
                print(
                    colored(f"Warning: symbol {symbol} already defined.
Address = {self.address}", color="yellow"))
        self.address += 1

    def print(self):
        print(tabulate([[x.symbol, x.address]
                        for x in self.fields], headers=["Symbol",
"Address"], tablefmt="psql"))
```

**symbol_field.py**

```
class SymbolField:
    def __init__(self, symbol, address):
        self.symbol = symbol
        self.address = address
```

**input file:**

```
    START 200
    MOVER AREG, ='5'
    MOVEM AREG, X
L1  MOVER BREG, ='2'
    ORIGIN L1 + 3
    LTORG
      ='50'
      ='10'
X   DS 1
    END
```

**Create and Display:**

```
(lab-2) → lab-2 git:(master) ✗ python index.py file.asm
Enter starting address: 200
Symbol: Address

Options:
  i: insert
  m: modify
  s: search
  d: display
  q: quit
  Your Choice: d
+----------+-----------+
| Symbol   |   Address |
|----------+-----------|
| L1       |       203 |
| X        |       208 |
+----------+-----------+

Options:
  i: insert
  m: modify
  s: search
  d: display
  q: quit
  Your Choice:
```

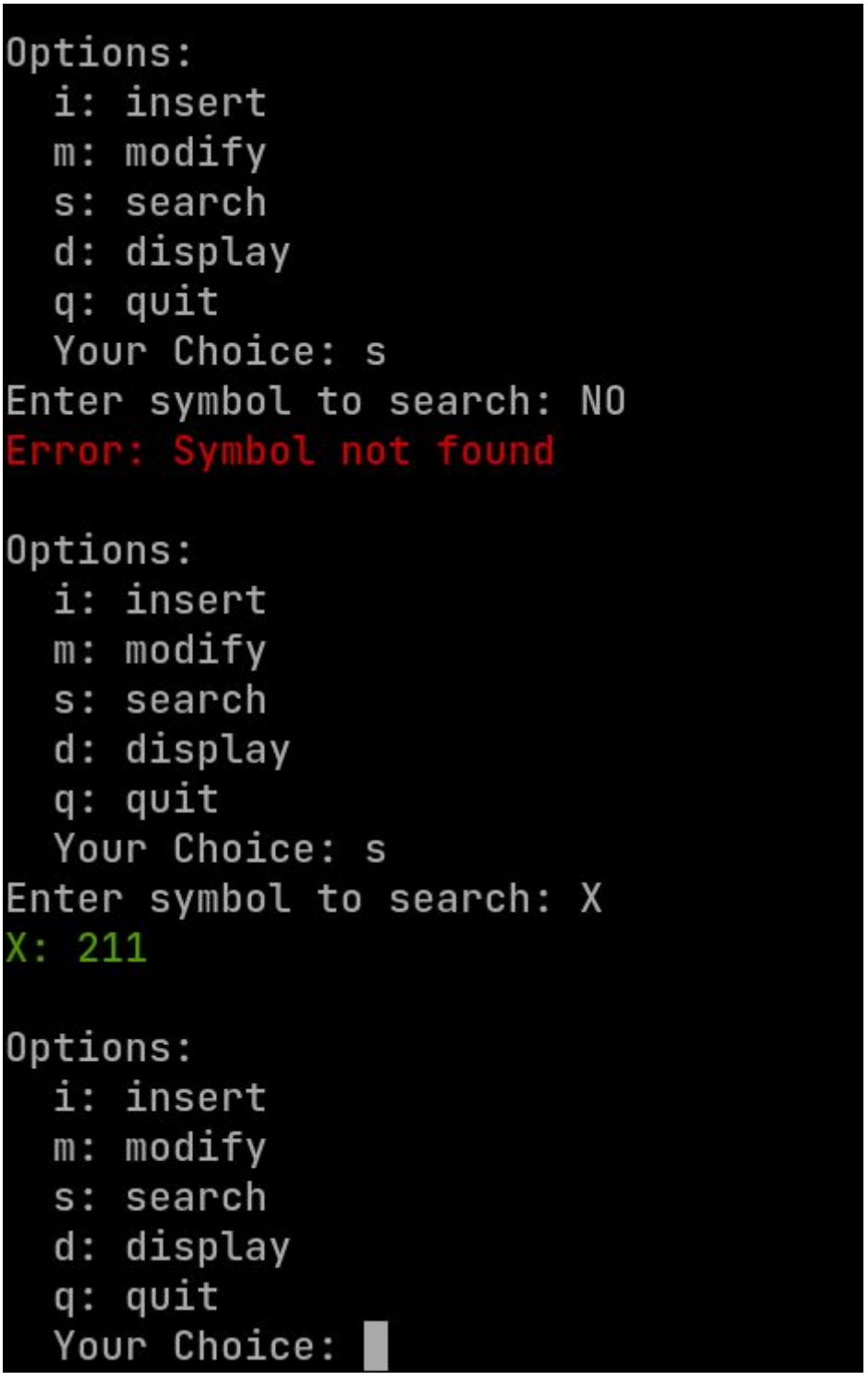
**insert:**

```
Options:
  i: insert
  m: modify
  s: search
  d: display
  q: quit
  Your Choice: i
Enter next instruction: UP MOVR AREG, ='5'
Inserted!

Options:
  i: insert
  m: modify
  s: search
  d: display
  q: quit
  Your Choice: d
+----------+-----------+
| Symbol   |   Address |
|----------+-----------|
| L1       |       203 |
| X        |       208 |
| UP       |       210 |
+----------+-----------+
```

**modify:**

```
Options:
  i: insert
  m: modify
  s: search
  d: display
  q: quit
  Your Choice: m
Enter symbol to modify: X
Enter new Address: 211
Modified!

Options:
  i: insert
  m: modify
  s: search
  d: display
  q: quit
  Your Choice: d
+----------+-----------+
| Symbol   |   Address |
|----------+-----------|
| L1       |       203 |
| X        |       211 |
| UP       |       210 |
+----------+-----------+
```

**search:**


Options:
i: insert
m: modify
s: search
d: display
q: quit
Your Choice: s
Enter symbol to search: NO
Error: Symbol not found
Options:
i: insert
m: modify
s: search
d: display
q: quit
Your Choice: s
Enter symbol to search: X
X: 211
Options:
i: insert
m: modify
s: search
d: display
q: quit
Your Choice: