**class AssemblerPass2:**

**def \_\_init\_\_(self):**

**self.opcode\_table={**

**"ADD" : "1",**

**"SUB" : "2",**

**"MUL" : "3",**

**"DIV":"4",**

**"LOAD": "5",**

**"STORE" : "6",**

**"START" : "00",**

**"END" : "FF"**

**}**

**self.symbol\_table = {}**

**self.literal= []**

**self.current\_add = 0**

**self.pool\_table = []**

**def pass\_two(self, intermediate\_code):**

**machine\_code =[]**

**for address, instruction in intermediate\_code:**

**if instruction in self.opcode\_table:**

**machine\_code.append((address,self.opcode\_table[instruction]))**

**elif instruction == "END":**

**machine\_code.append((address,"FF"))**

**elif instruction in self.symbol\_table:**

**symbol\_add = self.symbol\_table[instruction]**

**machine\_code.append((address,str(symbol\_add)))**

**elif instruction in self.literal:**

**machine\_code.append((address,instruction))**

**for index in self.pool\_table:**

**literal = self.literal[index]**

**machine\_code.append((self.current\_add,literal))**

**self.current\_add+=1**

**return machine\_code**

**def display\_machinecode(self,machine\_code):**

**for address,code in machine\_code:**

**print(f"{address}:{code}")**

**symbol\_table={**

**"A" :103,**

**"B":104,**

**"RESULT":105**

**}**

**literal = ["5","10"]**

**pool\_table =[0,1]**

**intermediate\_code = [**

**(100,"LOAD"),**

**(101,"ADD"),**

**(102,"STORE"),**

**(103,"A"),**

**(104,"B"),**

**(105,"RESULT"),**

**(106,"END")**

**]**

**ass=AssemblerPass2()**

**ass.symbol\_table = symbol\_table**

**ass.literal = literal**

**ass.pool\_table = pool\_table**

**machine\_code = ass.pass\_two(intermediate\_code)**

**ass.display\_machinecode(machine\_code)**