```
import java.io.*;
import java.util.*;

public class Pass2 {
	static Obj[] symb_table=new Obj[10];
	static Obj[] literal_table=new Obj[10];
	static int symb_found=0;

	public static void main(String[] args)throws IOException
	{
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter Total Number Of Symbols:");
		int total_symb=sc.nextInt();
		int pos,num;
		for(int i=0;i<total_symb;i++)
		{
			symb_table[i]=new Obj("",0);
			System.out.println("Enter Symbol Name:");
			symb_table[i].name=sc.next();
			System.out.println("Enter the Symbol Address");
			symb_table[i].addr=sc.nextInt();
		}
		System.out.println("Enter Total NUmber of Literals:");
		int total_ltr=sc.nextInt();
		for(int i=0;i<total_ltr;i++)
		{
			literal_table[i]=new Obj("",0);
			System.out.println("Enter Literal Name:");
			literal_table[i].name=sc.next();
			System.out.println("Enter the Literal Address");
			literal_table[i].addr=sc.nextInt();
		}

		System.out.println("\n*SYMBOL TABLE");
		System.out.println("\nSYMBOL\tADDRESS");
		for(int i=0;i<total_symb;i++)
			System.out.println(symb_table[i].name+"\t"+symb_table[i].addr);

		System.out.println("\n**LITERAL TABLE");
System.out.println("\nIndex\tLiteral\tAddress");
for (int i=0;i<total_ltr;i++)
        System.out.println(literal_table[i].name+"\t"+literal_table[i].addr);

BufferedReader br2=new BufferedReader(new FileReader("D:\\TCOB43\Assembler\\Output.txt"));
String line;
boolean symbol_error=false,undef_mnemonic=false;
System.out.println("\n*OUTPUT FILE*");
lab:while((line=br2.readLine())!=null)
{
        String[] token_list=line.split("\\s+",5);
        symbol_error=undef_mnemonic=false;
        lab1:for(String token:token_list)
        {
                if(token.matches("--"))
                {
                        System.out.println("\t--");
                        undef_mnemonic=true;
                }
                else if(token.matches("[0-9]+"))
                        System.out.println("\n\n"+token);
                else
                {
                        String letters=token.replaceAll("[A-Za-z]+", "");
                        num=Integer.parseInt(token.replaceAll("[0-9]+",""));

                        if(token.matches("\\[0-9]+\\)"))
                                System.out.println("\t"+num);
                        else
                        {
                                switch(letters.toUpperCase())
                                {
                                        case "S" : if(symb_table[num-1].addr == 0)
                                                {
                                                System.out.print("\t---");
                                                symbol_error = true;
                                                }
                                                else
                                                System.out.print("\t" + symb_table[num-
1].addr);
                                                                break;
                                                        case "L" : System.out.print("\t" + literal_table[num-
1].addr);
                                                        break;
                                                        case "AD" : System.out.print("\n");
                                                                continue lab;
                                                        case "DL" :
                                                                switch(num) {
                                                                        case 1:
                                                                        System.out.print("\n");
                                                                        continue lab;
                                                                        case 2: System.out.print("\t 00 \t
00");
                                                                }
                                                                continue lab1;
                                                        case "C" : System.out.print("\t" + num);
                                                                break;
                                                        default: System.out.print("\t" + "00" + num);
                                                }
                                        }
                                }
                        }
                }
                if(symbol_error)
                        System.out.println("\n\n*SYMBOL IS NOT DEFINED");
                if(undef_mnemonic)
                        System.out.print("\n\n**INVALID MNEMONIC*");
                int[] flag = new int[total_symb];
          for(int i=0;i<total_symb;i++){
            symb_found=0;
            for(int j=0;j<total_symb;j++){
              if(symb_table[i].name.equalsIgnoreCase(symb_table[j].name) && flag[j]==0){
                symb_found++;
                flag[i] = flag[j] = 1;
              }

            }
            if(symb_found>1){
              System.out.println("\n\n*"+symb_table[i].name+"IS DUPLICATE SYMBOL**");
```

```
                    }
                  }
                  br2.close();
                  sc.close();
                 }
}
```

Obj.java

```
public class Obj {
        String name;
        int addr;
        Obj(String nm,int address)
        {
        this.name=nm;
        this.addr=address;
        }
}
```

Pooltable.java

```
public class Pooltable {
        int first,total_literals;
        public Pooltable(int f,int l)
        {
                this.first=f;
                this.total_literals=l;
        }
}
```

OUTPUT:-

```
Enter Total Number Of Symbols:
5
Enter Symbol Name:
up
Enter the Symbol Address
102
Enter Symbol Name:
a
Enter the Symbol Address
109
Enter Symbol Name:
```

```
b
Enter the Symbol Address
121
Enter Symbol Name:
c
Enter the Symbol Address
122
Enter Symbol Name:
next
Enter the Symbol Address
102
Enter Total NUmber of Literals:
5
Enter Literal Name:
5
Enter the Literal Address
102
Enter Literal Name:
8
Enter the Literal Address
105
Enter Literal Name:
8
Enter the Literal Address
106
Enter Literal Name:
7
Enter the Literal Address
132
Enter Literal Name:
8
Enter the Literal Address
133
*SYMBOL TABLE
SYMBOL ADDRESS
up 102
```

a 109

b 121

c 122

next 102

**LITERAL TABLE

Index Literal Address

5 102

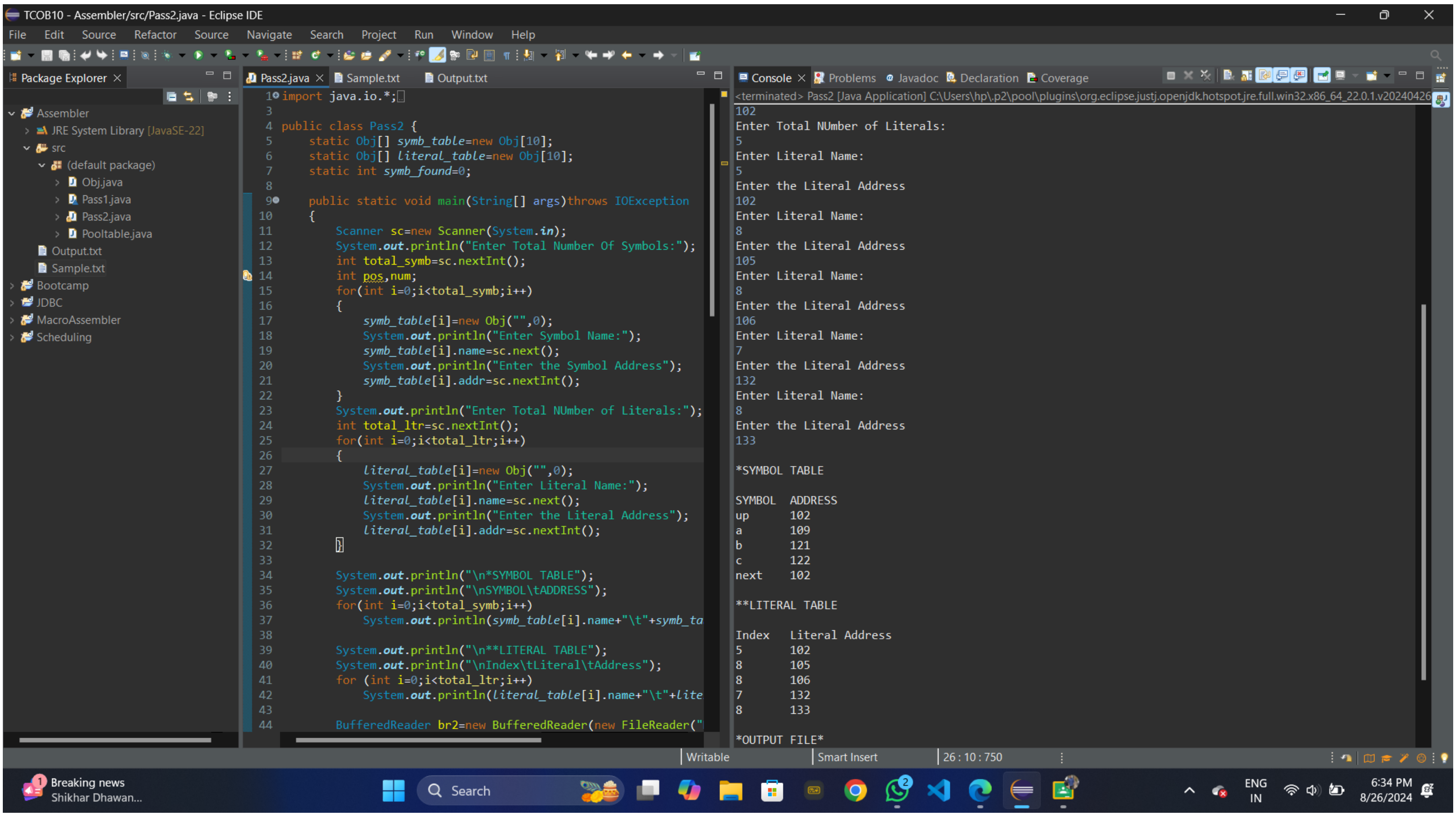
8 105

8 106

7 132

8 133

OUTPUT FILE



```java
import java.io.*;

public class Pass2 {
    static Obj[] symb_table=new Obj[10];
    static Obj[] literal_table=new Obj[10];
    static int symb_found=0;

    public static void main(String[] args)throws IOException
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter Total Number Of Symbols:");
        int total_symb=sc.nextInt();
        int pos,num;
        for(int i=0;i<total_symb;i++)
        {
            symb_table[i]=new Obj("",0);
            System.out.println("Enter Symbol Name:");
            symb_table[i].name=sc.next();
            System.out.println("Enter the Symbol Address");
            symb_table[i].addr=sc.nextInt();
        }
        System.out.println("Enter Total NUmber of Literals:");
        int total_ltr=sc.nextInt();
        for(int i=0;i<total_ltr;i++)
        {
            literal_table[i]=new Obj("",0);
            System.out.println("Enter Literal Name:");
            literal_table[i].name=sc.next();
            System.out.println("Enter the Literal Address");
            literal_table[i].addr=sc.nextInt();
        }

        System.out.println("\n*SYMBOL TABLE");
        System.out.println("\nSYMBOL\tADDRESS");
        for(int i=0;i<total_symb;i++)
            System.out.println(symb_table[i].name+"\t"+symb_ta

        System.out.println("\n**LITERAL TABLE");
        System.out.println("\nIndex\tLiteral\tAddress");
        for (int i=0;i<total_ltr;i++)
            System.out.println(literal_table[i].name+"\t"+lite

        BufferedReader br2=new BufferedReader(new FileReader("
```

```
<terminated> Pass2 [Java Application] C:\Users\hp\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v20240426
102
Enter Total NUmber of Literals:
5
Enter Literal Name:
5
Enter the Literal Address
102
Enter Literal Name:
8
Enter the Literal Address
105
Enter Literal Name:
8
Enter the Literal Address
106
Enter Literal Name:
7
Enter the Literal Address
132
Enter Literal Name:
8
Enter the Literal Address
133

*SYMBOL TABLE

SYMBOL  ADDRESS
up      102
a       109
b       121
c       122
next    102

**LITERAL TABLE

Index   Literal Address
5       102
8       105
8       106
7       132
8       133

*OUTPUT FILE*
```