INPUT:

```
import java.util.*;
import java.io.*;
public class pass1
{
       static int address=0;
       static int sadd[]=new int[10];
        static int ladd[]=new int[10];
        public static void main(String args[])
        {

                BufferedReader br;
                OutputStream oo;
                String input=null;

                String IS[]={"ADD","SUB","MUL","MOV"};
                String UserReg[]={"AREG","BREG","CREG","DREG"};
                String AD[]={"START","END"};
                String DL[]={"DC","DS"};
                int lc=0;
                int scount=0,lcount=0;
                int flag=0,flag2=0,stored=0;

                String tokens[]=new String[30];
                String tt=null;

                String sv[]=new String[10];
                String lv[]=new String[10];




                try
                {
                        br=new BufferedReader(new
FileReader("initial.txt"));
                        File f = new File("IM.txt");
                        File f1 = new File("ST.txt");
                        File f2 = new File("LT.txt");
                        PrintWriter p = new PrintWriter(f);
                        PrintWriter p1 = new PrintWriter(f1);
                        PrintWriter p2 = new PrintWriter(f2);
                     int k=0,l=0;
                        while ((input = br.readLine()) != null)
                        {
                                StringTokenizer st = new
StringTokenizer(input," ");
                                while (st.hasMoreTokens())
                                        {
                                             tt=st.nextToken();
```

```
//System.out.println(tt);

if(tt.matches("\\d*")&& tt.length() > 2)
                                                                {
lc=Integer.parseInt(tt);
p.println(lc);
address=lc-1;
                                                                }
                                                        else
                                                                {
       for(int i=0;i<AD.length;i++)
{
if(tt.equals(AD[i]))
{
p.print("AD "+(i+1)+" ");
}

}

for(int i=0;i<IS.length;i++)
                                                                        {
if(tt.equals(IS[i]))
{
p.print("IS "+(i+1)+" ");

}
}
for(int i=0;i<UserReg.length;i++)
                                                                        {
if(tt.equals(UserReg[i]))
```

```
{
p.print((i+1)+" ");
flag=1;
}

}
for(int i=0;i<DL.length;i++)                                                    {
if(tt.equals(DL[i]))
{
p.print("DL "+(i+1)+" ");
}
}
if(tt.length()==1 && !(st.hasMoreTokens()) && flag==1)
{

if ( Arrays.asList(sv).contains(tt) )
{

for(int i=0;i<scount;i++)
{


if(sv[i].equals(tt))
       {
              p.print("S"+i);
              flag2=1;
       }
```

```
        else
        {
                flag2=0;
        }
}

}
else
{
        p.print("S"+scount);
                sv[scount]=tt;
                flag2=1;
                scount++;
}

}

if(tt.length()==1 && (st.hasMoreTokens()))
{
                p.print(tt+" ");
                sadd[k]=address;k++;

}

if(tt.charAt(0)=='=')
{
                p.print("L"+lcount);
```

```
                lv[lcount]=tt;

                lcount++;

}

if(!st.hasMoreTokens())

{

                p.println();

}


if(tt.equals("DS"))
                                                                            {

        int a=Integer.parseInt(st.nextToken());

        address=address+a-1;

        p.println();
                                                                              }

                                                                        }



                                                         }
                                                        //System.out.println();
                                                        address++;
                                        } p.close();
                                        address--;

                                        for(int i=0;i<lcount;i++)
                                        {
                                            ladd[i]=address;
                                            address++;
                                        }



                                        for(int i=0;i<scount;i++)
                                        {
                                            p1.println(i+"\t"+sv[i]+"\t"+sadd[i]);
                                        }p1.close();

                                        for(int i=0;i<lcount;i++)
                                        {
```

```
                        p2.println(i+"\t"+lv[i]+"\t"+ladd[i]);
                    }p2.close();
                }
                catch(Exception e)
                {
                        e.printStackTrace();
                }    }  }
```

Initial.txt

```
START 100
MOV AREG A
MOV BREG B
MOV CREG =2
MOV DREG =3
ADD AREG BREG
SUB AREG A
A DC 05
B DS 03
END
```

OUTPUT:

IM.txt

```
AD 1 100
IS 4 1 S0
IS 4 2 S1
IS 4 3 L0
IS 4 4 L1
IS 1 1 2
IS 2 1 S0
A DL 1
B DL 2
AD 2
```

LT.txt

```
0       =2      110
1       =3      111
```

ST.txt

```
0       A       106
1       B       107
```