Pass 1 Macro INPUT :

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
public class spos3 {
	public static void main(String[] args) throws IOException {
		// TODO Auto-generated method stub
		BufferedReader br1=new BufferedReader(new FileReader("input.txt"));
		BufferedWriter bw=new BufferedWriter(new FileWriter("Output1.txt"));
		String line;
		mdt[] MDT=new mdt[20];
		mnt[] MNT=new mnt[4];
		arglist[] ARGLIST = new arglist[10];
		boolean macro_start=false,macro_end=false,fill_arglist=false;
		int mdt_cnt=0,mnt_cnt=0,arglist_cnt=0;
		while((line = br1.readLine())!=null)
		{
			line=line.replaceAll(",", " ");
			String[] tokens=line.split("\\s+");
			MDT[mdt_cnt] = new mdt();
			String stmnt = "";
			for(int i=0;i<tokens.length;i++)
			{
				if(tokens[i].equalsIgnoreCase("mend"))
				{
					MDT[mdt_cnt++].stmnt = "\t"+tokens[i];
					macro_end = true;
				}
				if(tokens[i].equalsIgnoreCase("macro"))
```

```
                {
                        macro_start = true;
                        macro_end = false;
                }
                else if(!macro_end)
                {
                        if(macro_start)
                        {
                                MNT[mnt_cnt++]=new mnt(tokens[i],mdt_cnt);
                                macro_start=false;
                                fill_arglist=true;
                        }
                        if(fill_arglist)
                        {
                                while(i<tokens.length)
                                {
                                        MDT[mdt_cnt].stmnt =
MDT[mdt_cnt].stmnt+ "\t" + tokens[i];
                                        stmnt = stmnt +"\t"+ tokens[i];
                                        if(tokens[i].matches("&[a-zA-
Z]+")||tokens[i].matches("&[a-zA-Z]+[0-9]+"))
                                                ARGLIST[arglist_cnt++]=new
arglist(tokens[i]);
                                        i++;
                                }
                                fill_arglist=false;
                        }
                        else
                        {
                                if(tokens[i].matches("[a-zA-Z]+") ||
tokens[i].matches("[a-zA-Z]+[0-9]+")||tokens[i].matches("[0-9]+"))
                                {
                                        MDT[mdt_cnt].stmnt =
MDT[mdt_cnt].stmnt+ "\t" + tokens[i];
```

```
                                        stmnt = stmnt +"\t"+ tokens[i];
                                    }
                                    if(tokens[i].matches("&[a-zA-Z]+") ||
tokens[i].matches("&[a-zA-Z]+[0-9]+"))
                                    {
                                        for(int j=0;j<arglist_cnt;j++)

        if(tokens[i].equals(ARGLIST[j].argname))
                                            {
                                                MDT[mdt_cnt].stmnt =
MDT[mdt_cnt].stmnt + "\t#"+(j+1);
                                                stmnt = stmnt
+"\t#"+(j+1);
                                            }
                                    }
                                }
                            }
                            else
                                bw.write(tokens[i]+"\t");
                        }
                        if(stmnt!="" && !macro_end)
                            mdt_cnt++;
                    }
                    br1.close();
                BufferedWriter bw1=new BufferedWriter(new FileWriter("MNT.txt"));
                    System.out.println("\n\t********MACRO NAME TABLE**********");
                    System.out.println("\n\tINDEX\tNAME\tADDRESS");
                    for(int i=0;i<mnt_cnt;i++)
                    {
                        System.out.println("\t"+i+"\t"+MNT[i].name+"\t"+MNT[i].addr);
                        bw1.write(MNT[i].name+"\t"+MNT[i].addr+"\n");
                    }
                    bw1.close();
```

```
                bw1=new BufferedWriter(new FileWriter("ARG.txt"));
                System.out.println("\n\n\t********ARGUMENT LIST**********");
                System.out.println("\n\tINDEX\tNAME\tADDRESS");
                for(int i=0;i<arglist_cnt;i++)
                {
                        System.out.println("\t"+i+"\t"+ARGLIST[i].argname);
                        bw1.write(ARGLIST[i].argname+"\n");
                }
                bw1.close();

                System.out.println("\n\t********MACRO DEFINITION
TABLE**********");
                System.out.println("\n\tINDEX\t\tSTATEMENT");

                bw1=new BufferedWriter(new FileWriter("MDT.txt"));
                for(int i=0;i<mdt_cnt;i++)
                {
                        System.out.println("\t"+i+"\t"+MDT[i].stmnt);
                        bw1.write(MDT[i].stmnt+"\n");
                }
                bw1.close();
        }
}
public class arglist {
        String argname;
        arglist(String argument) {
                // TODO Auto-generated constructor stub
                this.argname=argument;
        }
}
public class mnt {
```

```
        String name;
        int addr;
        int arg_cnt;
        mnt(String nm, int address)
        {
                this.name=nm;
                this.addr=address;
                this.arg_cnt=0;
        }
}
public class mdt {
String stmnt;
public mdt() {
        // TODO Auto-generated constructor stub
        stmnt="";
}
}
```

Pass 1 Macro INPUT FILE :

```
MACRO
INCR &X,&Y,&REG1 = AREG
MOVER &REG1,&X
ADD &REG1,&Y
MOVEM &REG1,&X
MEND
MACRO
DECR &A,&B,&REG2 = BREG
MOVER &REG2,&A
SUB &REG2,&B
MOVEM &REG2,&A
MEND
START 100
READ N1
READ N2
DECR N1,N2
INCR N1,N2
STOP
N1 DS 1
N2 DS 2
END
```

Pass 1 Macro OUTPUT :

********MACRO NAME TABLE**********

INDEX	NAME	ADDRESS
0	INCR	0
1	DECR	5

********ARGUMENT LIST**********

INDEX	NAME
0	&X
1	&Y
2	&REG1
3	&A
4	&B
5	&REG2

********MACRO DEFINITION TABLE**********

INDEX	STATEMENT				
0	INCR	&X	&Y	&REG1	= AREG
1	MOVER	#3	#1		
2	ADD	#3	#2		
3	MOVEM	#3	#1		
4	MEND				
5	DECR	&A	&B	&REG2	= BREG
6	MOVER	#6	#4		
7	SUB	#6	#5		
8	MOVEM	#6	#4		
9	MEND				

Pass 2 Macro INPUT :

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class spos4 {

	public static void main(String[] args) throws IOException {
		// TODO Auto-generated method stub
		mdt[] MDT=new mdt[20];
		mnt[] MNT=new mnt[4];
		arglist[] formal_parameter=new arglist[10];
		int macro_addr = -1;

		boolean macro_start=false,macro_end=false;
		int macro_call = -1;
		int
mdt_cnt=0,mnt_cnt=0,formal_arglist_cnt=0,actual_arglist_cnt=0,temp_cnt=0,temp_cnt1=0;

		BufferedReader br1=new BufferedReader(new FileReader("MNT.txt"));
		String line;
		while((line = br1.readLine())!=null)
		{
			String[] parts=line.split("\\s+");
			MNT[mnt_cnt++]=new mnt(parts[0],
Integer.parseInt(parts[1]),Integer.parseInt(parts[2]));
		}
		br1.close();
		System.out.println("\n\t********MACRO NAME TABLE**********");
```

```
		System.out.println("\n\tINDEX\tNAME\tADDRESS\tTOTAL
ARGUMENTS");
		for(int i=0;i<mnt_cnt;i++)

	System.out.println("\t"+i+"\t"+MNT[i].name+"\t"+MNT[i].addr+"\t\t"+MNT[i].arg_c
nt);


		br1=new BufferedReader(new FileReader("ARG.txt"));
		while((line = br1.readLine())!=null)
		{
			String[] parameters=line.split("\\s+");
			formal_parameter[formal_arglist_cnt++]=new arglist(parameters[0]);
			if(parameters.length>1)
				formal_parameter[formal_arglist_cnt-1].value = parameters[1];
		}
		br1.close();


		System.out.println("\n\n\t********FORMAL ARGUMENT
LIST**********");
		System.out.println("\n\tINDEX\tNAME\tADDRESS");
		for(int i=0;i<formal_arglist_cnt;i++)

	System.out.println("\t"+i+"\t"+formal_parameter[i].argname+"\t"+formal_parameter[i
].value);


		br1=new BufferedReader(new FileReader("MDT.txt"));
		while((line = br1.readLine())!=null)
		{
			MDT[mdt_cnt]=new mdt();
			MDT[mdt_cnt++].stmnt=line;
		}
		br1.close();
```

```
		System.out.println("\n\t********MACRO DEFINITION
TABLE**********");
		System.out.println("\n\tINDEX\t\tSTATEMENT");
		for(int i=0;i<mdt_cnt;i++)
			System.out.println("\t"+i+"\t"+MDT[i].stmnt);

		br1=new BufferedReader(new FileReader("input.txt"));
		arglist[] actual_parameter=new arglist[10];

		BufferedWriter bw1 = new BufferedWriter(new FileWriter("Output.txt"));
		while((line = br1.readLine())!=null)
		{
			line=line.replaceAll(",", " ");
			String[] tokens=line.split("\\s+");
			temp_cnt1=0;
			for(String current_token:tokens)
			{
				if(current_token.equalsIgnoreCase("macro"))
				{
					macro_start=true;
					macro_end=false;
				}
				if(macro_end && !macro_start)
				{
					if(macro_call != -1 && temp_cnt<formal_arglist_cnt-1)
					{
						if(formal_parameter[actual_arglist_cnt].value !=
"")

	actual_parameter[actual_arglist_cnt++]=new
arglist(formal_parameter[actual_arglist_cnt-1].value);


						actual_parameter[actual_arglist_cnt++]=new
arglist(current_token);
```

```
                                        if(formal_parameter[actual_arglist_cnt].value !=
""")

        actual_parameter[actual_arglist_cnt++]=new
arglist(formal_parameter[actual_arglist_cnt-1].value);


                                    }
                                    for(int i=0;i<mnt_cnt;i++)
                                    {
                                        if(current_token.equals(MNT[i].name))
                                        {
                                            macro_call=i;
                                            temp_cnt1 = temp_cnt1
+MNT[i].arg_cnt;
                                            break;
                                        }
                                        temp_cnt1 = temp_cnt1 + MNT[i].arg_cnt;
                                    }
                                    if(macro_call == -1)
                                        bw1.write("\t" + current_token);
                                }
                                if(current_token.equalsIgnoreCase("mend"))
                                {
                                    macro_end=true;
                                    macro_start=false;
                                }
                            }
                            if(macro_call != -1)
                            {
                                macro_addr=MNT[macro_call].addr+1;
                                while(true)
                                {
```

```
                                        if(MDT[macro_addr].stmnt.contains("mend") ||
MDT[macro_addr].stmnt.contains("MEND"))
                                        {
                                                macro_call = -1;
                                                break;
                                        }
                                        else
                                        {
                                                bw1.write("\n");
                                                String[]
temp_tokens=MDT[macro_addr++].stmnt.split("\\s+");
                                                for(String temp:temp_tokens)
                                                {
                                                        if(temp.matches("#[0-9]+"))
                                                        {
                                                                int num =
Integer.parseInt(temp.replaceAll("[^0-9]+", ""));

        bw1.write(actual_parameter[num-1].argname+"\t");
                                                        }
                                                        else
                                                                bw1.write(temp + "\t");
                                                }
                                        }
                                }
                        }
                        if(!macro_start )
                                bw1.write("\n");
                        macro_call= -1;
                }
                br1.close();
                bw1.close();
```

```
		System.out.println("\n\n\t********ACTUAL ARGUMENT
LIST**********");
		System.out.println("\n\tINDEX\tNAME\tADDRESS");
		for(int i=0;i<actual_arglist_cnt;i++)
			System.out.println("\t"+i+"\t"+actual_parameter[i].argname);
	}
}
public class arglist {
	String argname,value;
	arglist(String argument) {
		// TODO Auto-generated constructor stub
		this.argname=argument;
		this.value="";
	}
}public class mnt {
	String name;
	int addr;
	int arg_cnt;
	mnt(String nm, int address,int total_arg)
	{
		this.name=nm;
		this.addr=address;
		this.arg_cnt=total_arg;
	}
}
public class mdt {
String stmnt;
public mdt() {
	// TODO Auto-generated constructor stub
	stmnt="";
}
}
```

Pass 2 Macro INPUT FILE :

```
MACRO
INCR &X,&Y,&REG1
MOVER &REG1,&X
ADD &REG1,&Y
MOVEM &REG1,&X
MEND
MACRO
DECR &A,&B,&REG2
MOVER &REG2,&A
SUB &REG2,&B
MOVEM &REG2,&A
MEND
START 100
READ N1
READ N2
INCR N1,N2
DECR N1,N3
STOP
N1 DS 1
N2 DS 2
N3 DS 1
END
```

Pass 2 Macro OUTPUT :

********MACRO NAME TABLE**********

INDEX	NAME	ADDRESS	TOTAL ARGUMENTS
0	INCR	0	3
1	DECR	5	3

********FORMAL ARGUMENT LIST**********

INDEX	NAME	ADDRESS
0	&X	
1	&Y	
2	&REG1	AREG
3	&A	
4	&B	
5	&REG2	BREG

********MACRO DEFINITION TABLE**********

INDEX	STATEMENT					
0	INCR	&X	&Y	&REG1	=	AREG
1	MOVER		#3	#1		
2	ADD	#3	#2			
3	MOVEM		#3	#1		
4	MEND					
5	DECR	&A	&B	&REG2	=	BREG
6	MOVER		#6	#4		
7	SUB	#6	#5			
8	MOVEM		#6	#4		
9	MEND					

********ACTUAL ARGUMENT LIST**********

INDEX	NAMEADDRESS
0	N1
1	N2
2	AREG
3	N1
4	N3
5	BREG