Program:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Iterator;
import java.util.LinkedHashMap;

public class MacroPass1 {

        public static void main(String[] args) throws IOException{
                BufferedReader br=new BufferedReader(new
FileReader("macro_input.asm"));

                FileWriter mnt=new FileWriter("mnt.txt");
                FileWriter mdt=new FileWriter("mdt.txt");
                FileWriter kpdt=new FileWriter("kpdt.txt");
                FileWriter pnt=new FileWriter("pntab.txt");
                FileWriter ir=new FileWriter("intermediate.txt");
                LinkedHashMap<String, Integer> pntab=new LinkedHashMap<>();
                String line;
                String Macroname = null;
                int mdtp=1,kpdtp=0,paramNo=1,pp=0,kp=0,flag=0;
                while((line=br.readLine())!=null)
                {

                        String parts[]=line.split("\\s+");
                        if(parts[0].equalsIgnoreCase("MACRO"))
                        {
                                flag=1;
                                line=br.readLine();
                                parts=line.split("\\s+");
                                Macroname=parts[0];
                                if(parts.length<=1)
                                {

        mnt.write(parts[0]+"\t"+pp+"\t"+kp+"\t"+mdtp+"\t"+(kp==0?kpdtp:(kpdtp+1))+"\n");
                                        continue;
                                }
                                for(int i=1;i<parts.length;i++) //processing of parameters
                                {
                                        parts[i]=parts[i].replaceAll("[&,]", "");
                                        //System.out.println(parts[i]);
                                        if(parts[i].contains("="))
                                        {
                                                ++kp;
                                                String keywordParam[]=parts[i].split("=");
                                                pntab.put(keywordParam[0], paramNo++);
                                                if(keywordParam.length==2)
                                                {

        kpdt.write(keywordParam[0]+"\t"+keywordParam[1]+"\n");
                                                }
```

```
                    else
                    {
                        kpdt.write(keywordParam[0]+"\t-\n");
                    }
                }
                else
                {

                    pntab.put(parts[i], paramNo++);
                    pp++;
                }
            }

mnt.write(parts[0]+"\t"+pp+"\t"+kp+"\t"+mdtp+"\t"+(kp==0?kpdtp:(kpdtp+1))+"\n");
            kpdtp=kpdtp+kp;
            //System.out.println("KP="+kp);


        }
        else if(parts[0].equalsIgnoreCase("MEND"))
        {
            mdt.write(line+"\n");
            flag=kp=pp=0;
            mdtp++;
            paramNo=1;
            pnt.write(Macroname+":\t");
            Iterator<String> itr=pntab.keySet().iterator();
            while(itr.hasNext())
            {
                pnt.write(itr.next()+"\t");
            }
            pnt.write("\n");
            pntab.clear();
        }
        else if(flag==1)
        {
            for(int i=0;i<parts.length;i++)
            {
                if(parts[i].contains("&"))
                {
                    parts[i]=parts[i].replaceAll("[&,]", "");
                    mdt.write("(P,"+pntab.get(parts[i])+")\t");
                }
                else
                {
                    mdt.write(parts[i]+"\t");
                }
            }

            mdt.write("\n");
            mdtp++;
        }
        else
        {
            ir.write(line+"\n");
        }
    }
```

```
                br.close();
                mdt.close();
                mnt.close();
                ir.close();
                pnt.close();
                kpdt.close();
                System.out.println("Macro Pass1 Processing done. ....... )");
        }

}
```

Output:

```
Macro Pass1 Processing done........ )
```

Macro Pass1 file

```
--------------------
  Macro Input
--------------------
MACRO
M1      &X, &Y, &A=AREG, &B=
MOVER       &A, &X
ADD     &A, ='1'
MOVER       &B, &Y
ADD     &B,  ='5'
MEND
MACRO
M2      &P, &Q, &U=CREG, &V=DREG
MOVER       &U, &P
MOVER       &V, &Q
ADD     &U, ='15'
ADD     &V, ='10'
MEND
START       100
M1      10, 20, &B=CREG
M2      100, 200, &V=AREG, &U=BREG
END


--------------------
  Intermediate
--------------------
START       100
M1      10, 20, &B=CREG
M2      100, 200, &V=AREG, &U=BREG
END
```

```
---------------
  MDT
-----------------------
MOVER           (P,3)   (P,1)
ADD     (P,3)   ='1'
MOVER           (P,4)   (P,2)
ADD     (P,4)   ='5'
MEND
MOVER           (P,3)   (P,1)
MOVER           (P,4)   (P,2)
ADD     (P,3)   ='15'
ADD     (P,4)   ='10'
MEND

--------------------
  MNT
-----------------
M1      2       2       1       1
M2      2       2       6       3

--------------------
  PNTAB
---------------------
M1:     X       Y       A       B
M2:     P       Q       U       V
```

Pass 2:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.HashMap;
import java.util.Vector;


class MNTEntry {
	String name;
	int pp,kp,mdtp,kpdtp;
	public MNTEntry(String name, int pp, int kp, int mdtp, int kpdtp) {
		super();
		this.name = name;
		this.pp = pp;
		this.kp = kp;
		this.mdtp = mdtp;
		this.kpdtp = kpdtp;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public int getPp() {
		return pp;
	}
	public void setPp(int pp) {
		this.pp = pp;
	}
	public int getKp() {
		return kp;
	}
	public void setKp(int kp) {
		this.kp = kp;
	}
	public int getMdtp() {
		return mdtp;
	}
	public void setMdtp(int mdtp) {
		this.mdtp = mdtp;
	}
	public int getKpdtp() {
		return kpdtp;
	}
	public void setKpdtp(int kpdtp) {
		this.kpdtp = kpdtp;
	}
}


public class MacroPass2 {
	public static void main(String[] args) throws Exception {
		BufferedReader irb=new BufferedReader(new FileReader("intermediate.txt"));
		BufferedReader mdtb=new BufferedReader(new FileReader("mdt.txt"));
		BufferedReader kpdtb=new BufferedReader(new FileReader("kpdt.txt"));
		BufferedReader mntb=new BufferedReader(new FileReader("mnt.txt"));

		FileWriter fr=new FileWriter("pass2.txt");

		HashMap<String, MNTEntry> mnt=new HashMap<>();
		HashMap<Integer, String> aptab=new HashMap<>();
		HashMap<String,Integer> aptabInverse=new HashMap<>();

		Vector<String>mdt=new Vector<String>();
		Vector<String>kpdt=new Vector<String>();

		int pp,kp,mdtp,kpdtp,paramNo;
		String line;
		while((line=mdtb.readLine())!=null)
		{
			mdt.addElement(line);
		}
		while((line=kpdtb.readLine())!=null)
		{
			kpdt.addElement(line);
		}
		while((line=mntb.readLine())!=null)
		{
			String parts[]=line.split("\\s+");
			mnt.put(parts[0], new MNTEntry(parts[0], Integer.parseInt(parts[1]),
Integer.parseInt(parts[2]), Integer.parseInt(parts[3]), Integer.parseInt(parts[4])));

		}

		while((line=irb.readLine())!=null)
		{
			String []parts=line.split("\\s+");
			if(mnt.containsKey(parts[0]))
			{
				pp=mnt.get(parts[0]).getPp();
				kp=mnt.get(parts[0]).getKp();
				kpdtp=mnt.get(parts[0]).getKpdtp();
				mdtp=mnt.get(parts[0]).getMdtp();
				paramNo=1;
				for(int i=0;i<pp;i++)
				{
					parts[paramNo]=parts[paramNo].replace(",", "");
					aptab.put(paramNo, parts[paramNo]);
					aptabInverse.put(parts[paramNo], paramNo);
					paramNo++;
				}
				int j=kpdtp-1;
				for(int i=0;i<kp;i++)
				{
					String temp[]=kpdt.get(j).split("\t");
```

```
                    aptab.put(paramNo,temp[1]);
                    aptabInverse.put(temp[0],paramNo);
                    j++;
                    paramNo++;
                }

                for(int i=pp+1;i<parts.length;i++)
                {
                    parts[i]=parts[i].replace(",", "");
                    String splits[]=parts[i].split("=");
                    String name=splits[0].replaceAll("&", "");
                    aptab.put(aptabInverse.get(name),splits[1]);
                }
                int i=mdtp-1;
                while(!mdt.get(i).equalsIgnoreCase("MEND"))
                {
                    String splits[]=mdt.get(i).split("\\s+");
                    fr.write("+");
                    for(int k=0;k<splits.length;k++)
                    {
                        if(splits[k].contains("(P,"))
                        {
                            splits[k]=splits[k].replaceAll("[^0-9]",
"");//not containing number
                            String
value=aptab.get(Integer.parseInt(splits[k]));
                            fr.write(value+"\t");
                        }
                        else
                        {
                            fr.write(splits[k]+"\t");
                        }
                    }

                    fr.write("\n");
                    i++;
                }

                aptab.clear();
                aptabInverse.clear();
            }
            else
            {
                fr.write(line+"\n");
            }


    }

    fr.close();
    mntb.close();
    mdtb.close();
    kpdtb.close();
    irb.close();
    System.out.println("Macro Pass2 Processing done............. )");
    } }
```

Output:

```
Macro Pass2 Processing done .............)
```

Macro Pass2 file-

```
--------------------
   Macro Input
--------------------
  MACRO
  M1      &X, &Y, &A=AREG,
  &B=MOVER   &A, &X
  ADD    &A, ='1'
  MOVER          &B,
  &YADD          &B,
  ='5' MEND
  MACRO
  M2      &P, &Q, &U=CREG,
  &V=DREGMOVER  &U, &P
  MOVER          &V,
  &QADD          &U,
  ='15'
  ADD    &V,
  ='10'MEND
  START          100
  M1      10, 20, &B=CREG
  M2      100, 200, &V=AREG,
  &U=BREGEND

--------------------
   Intermediate
--------------------
  START          100
  M1      10, 20, &B=CREG
  M2      100, 200, &V=AREG,
  &U=BREGEND

--------------------
   Pass2
--------------------
  START          100
  +MOVER         AREG  10
  +ADD AREG  ='1'
  +MOVER         CREG  20
  +ADD CREG  ='5'
  +MOVER         BREG  100
  +MOVER         AREG  200
  +ADD BREG  ='15'
  +ADD  AREG
  ='10'END
```