Mpass2.java

```
import java.io.*;
import java.util.*;
public class Mpass2{
	public static void main(String[] args) throws IOException{
		mdt[] MDT=new mdt[20];
		mnt[] MNT=new mnt[10];
		arglist[] formal_parameter=new arglist[10];
		arglist[] actual_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("C:\\Users\\Vishal\\OneDrive\\Desktop\\SPOS\\MPass2\\src\\MNT.txt"));
		String line;
		while((line= br1.readLine()) !=null)
		{
			String[] parts=line.split("\\s+");
			//System.out.println("\t"+"\t"+parts[0]+"\t"+parts[1]+"\t\t"+parts[2]);
			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_cnt);
```

```
		br1=new BufferedReader(new
FileReader("C:\\Users\\Vishal\\OneDrive\\Desktop\\SPOS\\MPass2\\src\\ARGLIST.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\tVALUE");
		for(int i=0;i<formal_arglist_cnt;i++)

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

		br1=new BufferedReader(new
FileReader("C:\\Users\\Vishal\\OneDrive\\Desktop\\SPOS\\MPass2\\src\\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("C:\\Users\\Vishal\\OneDrive\\Desktop\\SPOS\\MPass2\\src\\input.txt"));

		BufferedWriter bw1=new BufferedWriter(new
FileWriter("C:\\Users\\Vishal\\OneDrive\\Desktop\\SPOS\\MPass2\\src\\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);
        }
}
```

---

mdt.java

```
public class mdt
{
        String stmnt;
        public mdt()
        {
                stmnt="";
        }
}
```

---

mnt.java

```
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;
        }
}
```

---

arglist.java

```
public class arglist{
        String argname,value;
        arglist(String argument){
```

```
                this.argname=argument;
                this.value="";
        }
        public arglist(String argname, String value) {
    this.argname = argname;
    this.value = value;
  }
}
```

ARGLIST.txt

```
&X
&Y
&REG1  AREG
&A
&B
&REG2  BREG
```

Input.txt

```
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
```

_____

MNT.txt

```
INCR    0       3

DECR    5       3
```

_____

MDT.txt

```
INCR    &X      &Y      &REG1 =      AREG

MOVER  #3       #1

ADD     #3      #2

MOVEM          #3      #1

MEND

DECR    &A      &B      &REG2 =      BREG

MOVER  #6       #4

SUB     #6      #5

MOVEM          #6      #4

MEND
```

_____

OUTPUT.txt

Keep it empty..output will generate here

Final output is

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

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

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

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

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

	INDEX					
t0	INCR	&X	&Y	&REG1	=	AREG
t1	MOVER	#3	#1			
t2	ADD	#3	#2			
t3	MOVEM		#3	#1		
t4	MEND					
t5	DECR	&A	&B	&REG2	=	BREG
t6	MOVER	#6	#4			
t7	SUB	#6	#5			
t8	MOVEM		#6	#4		
t9	MEND					

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

INDEX	NAME	ADDRESS
0	N1	
1	N2	
2	AREG	
3	N1.	
4	N3	
5	BREG	