CODE:

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

class M_Pass2
{
	public static void main(String[] args) throws IOException
	{
		File input = new File("m_intermediate.asm");
		input.createNewFile();
		File output = new File("m_output.asm");
		output.createNewFile();
		File tables = new File("m_tables.asm");
		tables.createNewFile();
		FileWriter fw = new FileWriter("m_output.asm");
		BufferedWriter bw = new BufferedWriter(fw);

		List<String> MDT = new ArrayList<String>();
		ArrayList<String[]> MNT = new ArrayList<String[]>();
		ArrayList<String[]> ALA = new ArrayList<String[]>();
		int mdtPtr = 0, alaPtr = 0;
		String[] tokens;

		//Reading tables from Pass 1
		Scanner fileReader = new Scanner(tables);
		int tableFlag = 0;
		int counter = 0;
		String[] a = new String[4];
		while(fileReader.hasNextLine())
		{
			String i_str = fileReader.nextLine();
			tokens = i_str.split("[ ,//n]");
			counter = 0;
			for(String str : tokens)
			{
				if(str.equals("[MDT]"))
				{
					tableFlag = 1;
					break;
				}
				else if(str.equals("[MNT]"))
				{
					tableFlag = 2;
					break;
				}
				else if(str.equals("[ALA]"))
				{
					tableFlag = 3;
					break;
				}
				switch(tableFlag)
				{
				case 2:
					a[counter++] = str;
					if(counter == 4)
					{
						MNT.add(new String[] {a[0],a[1],a[2],a[3]});
						counter = 0;
					}
					break;
```

```
                        case 3:
                                a[counter++] = str;
                                if(counter == 2)
                                {
                                        ALA.add(new String[] {a[0],a[1]});
                                        counter = 0;
                                }
                                break;
                        }
                }
                if(tableFlag == 1 && !i_str.equals("[MDT]"))
                        MDT.add(i_str);
        }
        fileReader.close();

        //Macroprocessor Pass 2
        fileReader = new Scanner(input);
        String[] newALA;
        while(fileReader.hasNextLine())
        {
                String i_str = fileReader.nextLine();
                String newstring = "";
                int CallCheckFlag = 0;    //0=Regular Code, 1=Macro Call
                tokens = i_str.split("[ ,//n]");
                CallCheckFlag = 0;
                String newline;
                for(String str : tokens)
                {
                        if(str.equals(""))
                                continue;
                        if(CallCheckFlag == 0)
                        {
                                for(String[] m : MNT)
                                {
                                        if(str.trim().equals(m[1]))      //Checks if token is in MNT
                                        {
                                                alaPtr = Integer.parseInt(m[2]);
                                                mdtPtr = Integer.parseInt(m[3])+1;
                                                CallCheckFlag = 1;
                                                break;
                                        }
                                }
                                if(CallCheckFlag == 0)    //Outputs non-Macro-name tokens
                                {
                                        newstring = newstring + str + " ";
                                }
                        }
                        else if(CallCheckFlag == 1)      //Sets Dummy Args to values from function
call
                        {
                                newALA = ALA.get(alaPtr);
                                newALA[1] = str;
                                ALA.set(alaPtr++, newALA);
                        }
                }
                while(CallCheckFlag == 1) //Expanding Macro
                {
                        tokens = MDT.get(mdtPtr++).split("[ ,//n]");
                        newline = "";
                        for(String str : tokens)
                        {
```

```
                                    if(str.charAt(0) == '#')  //Inserts Actual Arguments
                                    {
                                          newline = newline +
ALA.get(Integer.parseInt(str.substring(1,str.length())))[1] + " ";
                                    }
                                    else if(str.equals("MEND"))
                                          CallCheckFlag = 0;
                                    else
                                    {
                                          newline = newline + str + " ";
                                    }
                              }
                              newstring = newstring + "\t" + newline.trim();
                              if(CallCheckFlag != 0)
                                    newstring = newstring + "\n";
                        }
                        if(newstring != "")
                        {
                              if(newstring.charAt(0)=='\t')
                                    newstring = "\t" + newstring.trim();
                              bw.write(newstring);
                              if(newstring.charAt(newstring.length()-1) != '\n')
                                    bw.write("\n");
                        }
                  }

                  fileReader.close();

                  bw.close();
                  System.out.println("MDT: " + MDT);
                  System.out.println("\nMNT: ");
                  for(String[] arr : MNT)
                        System.out.println(Arrays.toString(arr));
                  System.out.println("\nALA: ");
                  for(String[] arr : ALA)
                        System.out.println(Arrays.toString(arr));

                  fileReader = new Scanner(output);
                  System.out.println("\n\nFinal Code");
                  while(fileReader.hasNextLine())
                  {
                        System.out.println(fileReader.nextLine());
                  }
                  fileReader.close();
            }
}
```

INPUT:

m_intermediate.asm

```
      START
            MOVER AREG S1
            MOVER BREG S1
            INCR D1 D2
      S1    DC 5
      D1    DC 2
      D2    DC 3
      END
```

m_tables.asm

```
        [MDT]
        INCR &ARG1 &ARG2
        ADD AREG #0
        ADD BREG #1
        MEND
        [MNT]
        0 INCR 0 0
        [ALA]
        0 &ARG1
        1 &ARG2
```

OUTPUT: