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

class MntTuple {
  String name;
  int index;

  MntTuple(String s, int i) {
    name = s;
    index = i;
  }

  public String toString() {
    return("[" + name + ", " + index + "]");
  }
}

class MacroProcessor{
  static List<MntTuple> mnt;
  static List<String> mdt;
  static int mntc;
  static int mdtc;
  static int mdtp;
  static BufferedReader input;
  static List<List <String>> ala;
  static Map<String, Integer> ala_macro_binding;

  public static void main(String args[]) throws Exception {
    initializeTables();
    System.out.println("===== PASS 1 =====\n");
    pass1();
  }

  static void pass1() throws Exception {
    String s = new String();
    input = new BufferedReader(new InputStreamReader(new
FileInputStream("input.txt")));
    PrintWriter output = new PrintWriter(new
FileOutputStream("output_pass1.txt"), true);
    while((s = input.readLine()) != null) {
      if(s.equalsIgnoreCase("MACRO")) {
        processMacroDefinition();
      } else {
        output.println(s);
      }
    }
    System.out.println("ALA:");
    showAla(1);
    System.out.println("\nMNT:");
    showMnt();
    System.out.println("\nMDT:");
    showMdt();
  }

  static void processMacroDefinition() throws Exception {
    String s = input.readLine();
    String macro_name = s.substring(0, s.indexOf(" "));
    mnt.add(new MntTuple(macro_name, mdtc));
    mntc++;
    pass1Ala(s);
    StringTokenizer st = new StringTokenizer(s, " ,", false);
    String x = st.nextToken();
    for(int i=x.length() ; i<12 ; i++) {
      x += " ";
    }
    String token = new String();
    int index;
    token = st.nextToken();
    x += token;
    while(st.hasMoreTokens()) {
      token = st.nextToken();
      x += "," + token;
    }
    mdt.add(x);
    mdtc++;
    addIntoMdt(ala.size()-1);
  }

  static void pass1Ala(String s) {
    StringTokenizer st = new StringTokenizer(s, " ,", false);
    String macro_name = st.nextToken();
    List<String> l = new ArrayList<>();
    int index;
    while(st.hasMoreTokens()) {
      String x = st.nextToken();
      if((index = x.indexOf("=")) != -1) {
        x = x.substring(0, index);
      }
      l.add(x);
    }
    ala.add(l);
    ala_macro_binding.put(macro_name, ala_macro_binding.size());
  }

  static void addIntoMdt(int ala_number) throws Exception {
    String temp = new String();
    String s = new String();
    List l = ala.get(ala_number);
    boolean isFirst;
    while(!s.equalsIgnoreCase("MEND")) {
      isFirst = true;
      s = input.readLine();
      String line = new String();
      StringTokenizer st = new StringTokenizer(s, " ,", false);
      temp = st.nextToken();
      for(int i=temp.length() ; i<12 ; i++) {
        temp += " ";
      }
      line += temp;
      while(st.hasMoreTokens()) {
        temp = st.nextToken();
        if(temp.startsWith("&")) {
          int x = l.indexOf(temp);
          temp = ",#" + x;
          isFirst = false;
        } else if(!isFirst) {
          temp = "," + temp;
        }
        line += temp;
      }
      mdt.add(line);
      mdtc++;
    }
  }

  static void showAla(int pass) throws Exception {
    PrintWriter out = new PrintWriter(new FileOutputStream("out_ala_pass" + pass + ".txt"), true);
    for(List l : ala) {
      System.out.println(l);
      out.println(l);
    }
  }

  static void showMnt() throws Exception {
    PrintWriter out = new PrintWriter(new FileOutputStream("out_mnt.txt"), true);
    for(MntTuple l : mnt) {
      System.out.println(l);
      out.println(l);
    }
  }

  static void showMdt() throws Exception {
    PrintWriter out = new PrintWriter(new FileOutputStream("out_mdt.txt"), true);
    for(String l : mdt) {
      System.out.println(l);
      out.println(l);
    }
  }
  static void initializeTables() {
    mnt = new LinkedList<>();
    mdt = new ArrayList<>();
    ala = new LinkedList<>();
    mntc = 0;
    mdtc = 0;
    ala_macro_binding = new HashMap<>();
  }
}
```