```
import java_cup.runtime.*;
import java.util.ArrayList;

terminal ALL, CLL, AP, CP, AC, CC;
terminal COMA, PYC, ASIG, MAS, MENOS, POR, DIV;
terminal INVERSA, TRANSPUESTA, ADJUNTA, PRINT;
terminal String IDENT;
terminal Double NUMERO;

non terminal linea, lineaExp;
non terminal double[][] comando;
non terminal double[][] matriz, exp;
non terminal ArrayList<ArrayList<Double>> procesarMatriz;
non terminal ArrayList<Double> fila;

precedence left MAS, MENOS;
precedence left POR, DIV;

lineaExp ::= lineaExp linea | linea;

linea ::= exp:v PYC | PYC;

exp ::= PRINT AP exp:v {:Matrices.print(v);:} CP
      | INVERSA AP exp:v {:if(Matrices.filas(v)==Matrices.columnas(v)){
                              RESULT = Matrices.inversa(v);
                           }else{
                              System.out.println(Matrices.ERROR_INVERSA);
                              System.exit(-1);
                           }:} CP
      | TRANSPUESTA AP exp:v {:RESULT = Matrices.transpuesta(v);:} CP
      | ADJUNTA AP exp:v {:if(Matrices.filas(v)==Matrices.columnas(v)){
                              RESULT = Matrices.adjunta(v);
                           }else{
                              System.out.println(Matrices.ERROR_ADJUNTA);
                              System.exit(-1);
                           }:} CP
      | exp:v1 MAS exp:v2 {:if((Matrices.columnas(v1)==Matrices.columnas(v2))&&
(Matrices.filas(v1)==Matrices.filas(v2))){
                              RESULT = Matrices.suma(v1,v2);
                            }else{
                              System.out.println(Matrices.ERROR_SUMA);
                              System.exit(-1);
                            }:}
      | exp:v1 MENOS exp:v2
      | exp:v1 POR exp:v2 {:if(Matrices.columnas(v1)==Matrices.filas(v2)){
                              RESULT = Matrices.producto(v1,v2);
                            }else{
                              System.out.println(Matrices.ERROR_PROD);
                              System.exit(-1);
                            };:}
      | exp:v1 DIV exp:v2
      | IDENT:a ASIG exp:v {:TablaSimbolos.insertar(a, v);:}
      | IDENT:a {:double[][] aux = TablaSimbolos.buscar(a);
                  if(aux==null){
                        System.out.println(TablaSimbolos.ERROR_NOEXISTE);
                        System.exit(-1);
                  }
                  RESULT = aux;:}
      | AP exp:v CP {:RESULT=v;:}
      | matriz:v {:RESULT = v;:};


matriz ::= AC procesarMatriz:v CC {:try{
                                          double[][] a = Matrices.toArray(v);
                                          RESULT=a;
                                    }catch (IndexOutOfBoundsException e){
                                          System.out.println(Matrices.ERROR_FILAS);
                                          System.exit(-1);
                                    }
                                    :}
         |ALL procesarMatriz:v CLL {:try{
                                          double[][] a = Matrices.toArray(v);
                                          RESULT=a;
                                    }catch (IndexOutOfBoundsException e){
                                          System.out.println(Matrices.ERROR_FILAS);
                                          System.exit(-1);
                                    }
                                    :};


procesarMatriz ::= fila:v PYC procesarMatriz:m {:ArrayList<ArrayList<Double>> aux = new ArrayList<>(); aux.add(v);aux.addAll(m); RESULT = aux;:}
                 | fila:v {:ArrayList<ArrayList<Double>> aux = new ArrayList<>(); aux.add(v); RESULT = aux;:}
                 | ALL fila:v CLL COMA procesarMatriz:m {:ArrayList<ArrayList<Double>> aux = new ArrayList<>(); aux.add(v);aux.addAll(m); RESULT = aux;:}
                 | ALL fila:v CLL {:ArrayList<ArrayList<Double>> aux = new ArrayList<>(); aux.add(v); RESULT = aux;:} ;

fila ::= NUMERO:a COMA fila:v {:ArrayList<Double> aux = new ArrayList<>(); aux.add(a); aux.addAll(v); RESULT=aux;:}
       | NUMERO:a {:ArrayList<Double> aux = new ArrayList<>(); aux.add(a); RESULT = aux;:};

```