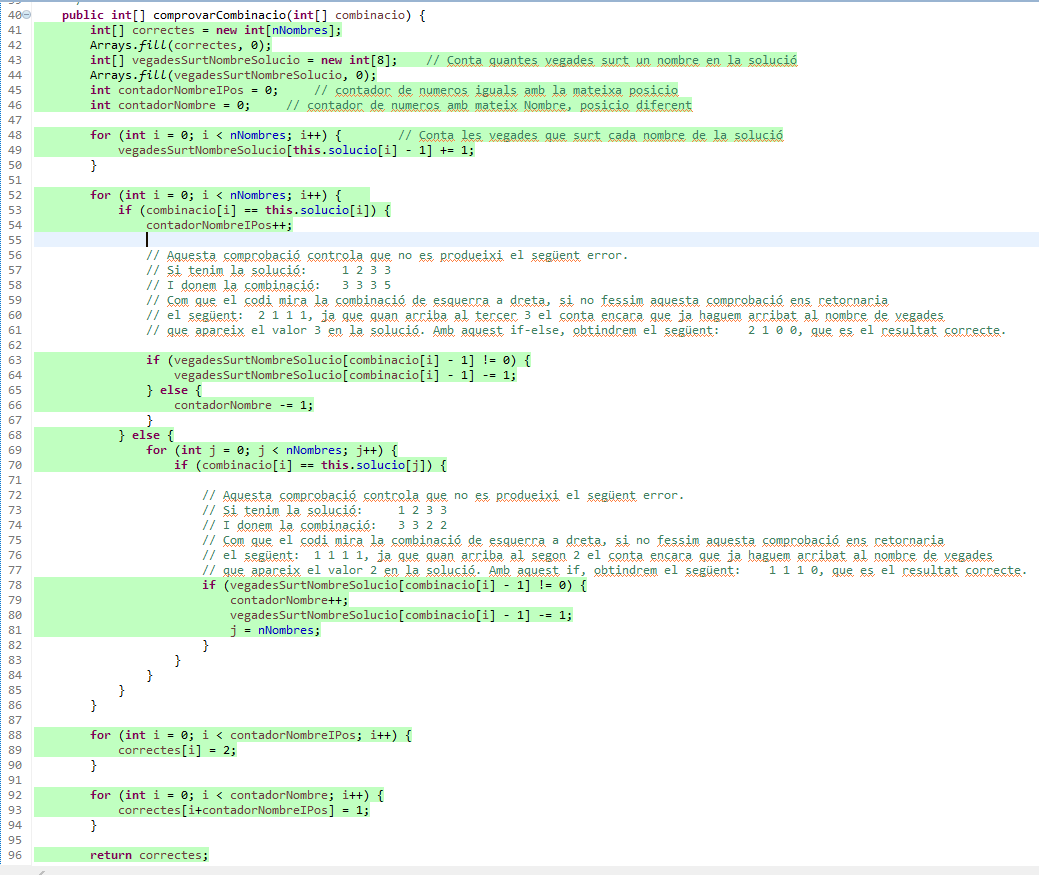
**Proves de caixa blanca**

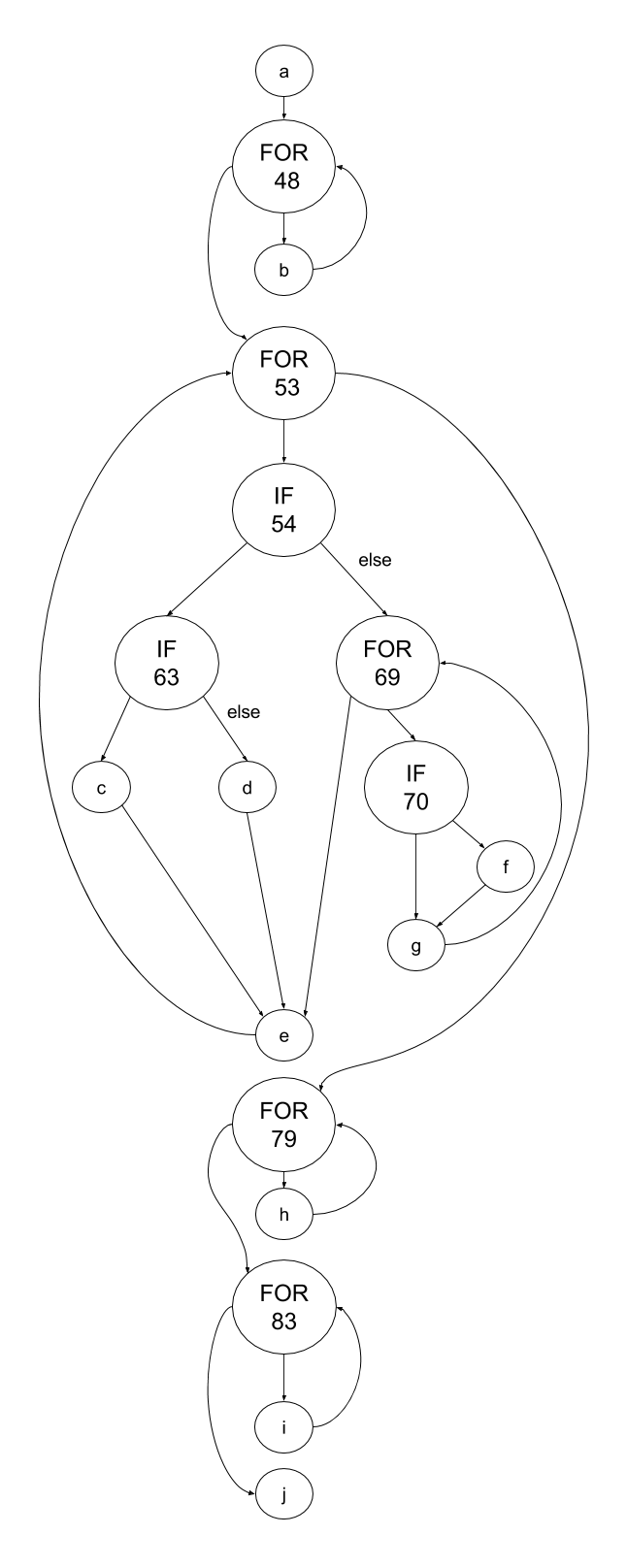
En el nostre codi, hem utilitzat les proves de caixa blanca per testejar la funció comprovarCombinacio() de la classe Solució, ja que es la única funció del programa que té una certa complexitat i té sentit realitzar aquests tests.

* **Control flow**: s’ha controlat el flow amb el eclemma.
* **Statement coverage**: totes les sentencies de la funció s’executen com a mínim una vegada.

Statement coverage de la funció comprovarCombinacio de Solucio.java

* **Decision coverage**: totes les decisions prenen el valor true i false al menys una vegada cadascun. Aquest test es realitza mitjançant tots els test de la funció, ja que amb les diferents combinacions testejades ja es compleixen totes les condicions.
* **Condition coverage**: les conditions de la funció només tenen una decisió, així que pel punt anterior ja es compleix el Condition coverage.
* **Loop testing**: hem agafat els bucles de la funció i els hem fet recórrer per diferents valors:
  + **Bucles nNombres** (for-48, for-53 i for-69): des de 0 fins a 6 (ja que aquest és el nombre màxim de xifres que pot tenir una combinació en el nivell de dificultat Difícil).
  + **Bucle contadorNombreIPos** (for-88):des de 0 fins a 6, igual que per els bucles anteriors.
  + **Bucle contadorNombre** (for-92): des de 0 fins a 6, igual que per els bucles anteriors.
* **Path coverage**: les proves realitzades cobreixen tots els paths independents de la funció.

1. a, for-48, b, for-48, for-53, if-54, if-63, c, e, for-53, for-88, h, for-88, for-92, j (**test comb\_0**).
2. a, for-48, b, for-48, for-53, if-54, if-63, d, e, for-53, for-88, h, for-88, for-92, i, for-92, j (**test comb\_14**).
3. a, for-48, b, for-48, for-53, if-54, for-69, if-70, g, for-69, e, for-53, for-88, for-92, j (**test comb\_2**).
4. a, for-48, b, for-48, for-53, if-54, for-69, if-70, f, g, for-69, e, for-53, for-88, for-92, i, for-92, j (**test comb\_1**)



Graf de fluxe de la funció comprovarCombinacio de Solucio.java