

#### ***EJERCICIO 4.- Compilación (1,5 puntos)***

En cualquier lenguaje de programación existen relaciones entre programas de modo que un programa PR1 utiliza otros programas PR2, PR3 y a su vez estos pueden necesitar de otros programas PR4, PR5, ... Con tantas llamadas a programas, un compilador necesita saber, dado un programa, si están definidos todos los programas que utiliza tanto directa como indirectamente.

Para almacenar las relaciones entre programas utilizaremos una tabla hash en la cual se guarda, utilizando como clave el nombre del programa, la **lista** de nombres de los programas utilizados por éste.

***Tabla hash***

<b><i>Clave</i></b>	<b><i>Datos (Lista de Programas utilizados )</i></b>
<b>PR1</b>	<b>PR2, PR3</b>
<b>PR2</b>	<b>PR5, PR3</b>
<b>PR15</b>	<b>PR5, PR14</b>
<b>PR3</b>	<b>PR5</b>
<b>PR5</b>	<b>PR2</b>
<b>PR14</b>	<b>PR25</b>

Por ejemplo, el subprograma PR1 es válido porque se encuentra definido en la tabla, así como los subprogramas PR2, PR3 y PR5 que utiliza (directa o indirectamente). Del mismo modo, el subprograma PR15 no es válido porque hace uso indirecto de PR25, que no se encuentra definido como clave en la tabla.

Se pide **implementar** el subprograma *esVálido(programa)* que dirá si un programa puede compilarse correctamente, **calculando su coste**.

```
public class Compilador {  
  
    HashMap<String, ArrayList<String>> dependencias;  
  
    public boolean esVálido(String prog)  
    // Postcondición: el resultado será true si se han definido TODOS los  
    //                programas usados directa o indirectamente por "prog" y false si no  
  
}
```

#### **ARIKETA 4.- Konpilazioa (1,5 puntu)**

Programazio-lengoaia guztietan programen arteko erlazioak egoten dira. Era horretan, PR1 programa batek PR2, PR3 programak erabiltzen ditu, eta hauek PR4, PR5 ... programak erabil ditzakete. Hainbeste erlazioarekin, konpiladore batek jakin behar du ea, programa bat emanda, berak erabiltzen dituen programa guztiak definituta dauden (zuzenean eta zeharka ere).

Programen arteko erlazioak gordetzeko hash taula bat erabiliko dugu, gakoa programaren izena delarik, eta datua programa horrek erabiltzen dituen programen zerrenda.

##### ***hash taula***

**Gakoa**                      **Datuak (erabilitako programen zerrenda)**

PR1	PR2, PR3
PR2	PR5, PR3
PR15	PR5, PR14
PR3	PR5
PR5	PR2
PR14	PR25

Adibidez, PR1 programa zuzena da, taulan definituta dagoelako eta, gainera, berak erabiltzen dituen PR2, PR3 eta PR5 programak ere definituta daudelako (zuzenean edo beste programa batzuen bitartez). Era berean, PR15 ez da zuzena, zeharka PR25 erabiltzen duelako, eta programa hori ez dago definituta hash taulan.

*zuzenaDa(programa)* **implementatu** behar da, programa hori ondo konpilatu daitekeen jakiteko, eta bere **kostua kalkulatu**.

```
public class Konpiladorea {  
    HashMap<String, ArrayList<String>> mendekotasunak;  
  
    public boolean zuzenaDa(String prog)  
    // Postbaldintza: emaitza true "prog" programak erabiltzen dituen programa GUZTIAK  
    //                               definituta baldin badaude, eta false bestela  
  
}
```