

## Ejercicio 1

```
problema ultima_aparicion (s: seq Z , e: Z ) : Z {  
  requiere: {e pertenece a s }  
  asegura: {res es la posición de la última aparición de e en s}  
}
```

## Ejercicio 2

```
problema elementos_exclusivos (s: seq Z , t: seq Z ) : seq Z {  
  requiere: -  
  asegura: {Los elementos de res pertenecen o bien a s o bien a t, pero no a ambas }  
  asegura: {res no tiene elementos repetidos }  
}
```

## Ejercicio 3

Se cuenta con un diccionario que contiene traducciones de palabras del idioma castellano (claves) a palabras en inglés (valores), y otro diccionario que contiene traducciones de palabras en castellano (claves) a palabras en alemán (valores). Se pide escribir un programa que dados estos dos diccionarios devuelva la cantidad de palabras que tienen la misma traducción en inglés y en alemán.

```
problema contar_traduccion_iguales (ing: dict String,String , ale: dict String,String ) : Z {  
  requiere: -  
  asegura: {res = cantidad de palabras que están en ambos diccionarios y además tienen igual valor en ambos}  
}
```

## Ejercicio 4

Dada una lista de enteros s, se desea devolver un diccionario cuyas claves sean los valores presentes en s, y sus valores la cantidad de veces que cada uno de esos números aparece en s

```
problema convertir_a_diccionario (lista: seq Z ) : dict Z,Z {  
  requiere: -  
  asegura: {res tiene como claves los elementos de lista y res[n] = cantidad de veces que aparece n en lista}  
}
```