

Introducción a la Programación

Segundo Parcial - Simulacro

- El parcial se aprueba con 6 puntos
 - Utilizar [este](#) archivo fuente de base para la programación. Ya cuenta con los def y las signatures correctas.
 - Para testear el código pueden usar [este](#) archivo que ya cuenta con todo lo necesario para desarrollar sus propios tests (este archivo no se entrega)
 - **Para aprobar el parcial es requisito indispensable que todos los programas pasen los tests del archivo del punto anterior**
-

1) Última aparición

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

TIP: realizar la iteración mediante índices y no mediante elementos

Por ejemplo, dados
 $s = [-1, 4, 0, 4, 100, 0, 100, 0, -1, -1]$
 $e = 0$
se debería devolver $res = 7$

2) Elementos exclusivos

problema elementos_exclusivos ($s: \text{seq}\langle Z \rangle$, $t: \text{seq}\langle Z \rangle$) : $\text{seq}\langle Z \rangle$ {
requiere: -
asegura: { res tiene todos los elementos que pertenecen o bien a s o bien a t , pero no a ambas }
asegura: { res no tiene elementos repetidos }
}

Por ejemplo, dados
 $s = [-1, 4, 0, 4, 3, 0, 100, 0, -1, -1]$
 $t = [0, 100, 5, 0, 100, -1, 5]$
se debería devolver $res = [3, 4, 5]$ ó $res = [3, 5, 4]$ ó $res = [4, 3, 5]$ ó $res = [4, 5, 3]$ ó $res = [5, 3, 4]$ ó $res = [5, 4, 3]$

3) Contar traducciones iguales

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

Por ejemplo, dados los diccionarios

aleman = {"Mano": "Hand", "Pie": "Fuss", "Dedo": "Finger", "Cara": "Gesicht"}

inglés = {"Pie": "Foot", "Dedo": "Finger", "Mano": "Hand"}

se debería devolver *res* = 2

4) Convertir a diccionario

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>) : dicc<Z,Z> {  
  requiere: -  
  asegura: {res tiene como claves los elementos de lista y res[n] = cantidad de veces que aparece n en lista}  
}
```

Por ejemplo, dada la lista

lista = [-1, 0, 4, 100, 100, -1, -1]

se debería devolver *res* = {-1:3, 0:1, 4:1, 100:2}

RECORDAR QUE NO IMPORTA EL ORDEN DE LAS CLAVES EN UN DICCIONARIO