

Descripción del caso	a>0	b>0	c>0	Valor	a > b+c	b > a+c	c > a +b	Resultado esperado	Comentario
a menor a 0	Sí	-	-	-	-	-	-	ERROR	-
b menor a 0	-	Sí	-	-	-	-	-	ERROR	-
c menor a 0	-	-	Sí	-	-	-	-	ERROR	-
Todos son iguales	Sí	Sí	Sí	a = b = c	Sí	Sí	Sí	Equilátero	-
a y b son iguales y c es mayor a (a+b)	Sí	Sí	Sí	a=b y c /= a,b	No	No	Sí	Isósceles	-
b y c son iguales y a es mayor a (b+c)	Sí	Sí	Sí	b = c y a/= b,c	Sí	No	No	Isósceles	-
a y c son iguales y b es mayor a (a+c)	Sí	Sí	Sí	a=c y b/=a,c	No	Sí	No	Isósceles	-
c mayor a a+b	Sí	Sí	Sí	a /= b, a/=c, b/=c	-	-	Sí	Escaleno	-
a mayor a b+c	Sí	Sí	Sí	a /= b, a/=c, b/=c	Sí	-	-	Escaleno	-
b mayor a a+c	Sí	Sí	Sí	a /= b, a/=c, b/=c	-	Sí	-	Escaleno	-
a menor o igual a (b+c) y b menor o igual a (a+c) y c menor o igual a (a+b)	-	-	-	-	NO	NO	NO	ERROR	-

problema parteEntera (x: R) : Z{ requiere: { True } asegura: { resultado $\leq$ x < resultado + 1 }}					
Método de partición por categorías					
1) Problemas a testear		7) Armar los casos de test			
parteEntera		Descripción del caso	¿Es entero?	Resultado esperado	Datos de test
2) Elegir una unidad funcional		x no tiene decimales	Sí	x	2
parteEntera		x tiene decimales	No	x redondeado a uno de sus dos enteros más cercanos	2.3
3) Identificar factores					
x: R					
4) Determinar categorías					
x:R - ¿es entero?					
5) Determinar elecciones					
x: R - ¿es entero?					
-Sí					
-No					
6) Clasificar elecciones					
x: R - ¿es entero?					
-Sí [Único]					
-No [Único]					

problema quitarTodos (e: T , s: seq(T )) : seq(T ) { requiere: { True } asegura: { resultado es igual a s pero sin el elemento e. } }						
Método de partición por categorías						
1) Problemas a testear		7) Armar los casos de test				
quitarTodos		Descripción del caso	¿e está en s?	¿s tiene elementos?	Resultado esperado	Datos de test
2) Elegir una unidad funcional		secuencia vacía	No	No	secuencia vacía	s = [ ], e= 2
quitarTodos		secuencia sin e	No	Sí	s	s = [1,2,3,4], e=5
3) Identificar factores		secuencia con e	Sí	Sí	s sin e	s = [1,2,3,4], e=2
e: T , s: seq(T)						
4) Determinar categorías						
e:T -¿e está en s?						
s:seq(T) - ¿Tiene elementos?						
5) Determinar elecciones						
e: T - ¿Está en s?	s:seq(T) -¿Tiene elementos?					
-Sí	-Sí					
-No	-No					
6) Clasificar elecciones						
e: T - ¿Está en s?	s:seq(T) -¿Tiene elementos?					
-Sí [Único]	-Sí [Restricción]					
-No [Restricción]	-No [Único]					

problema sumarN (n: Z, s: seq(Z)) : seq(Z) { requiere: { True } asegura: { resultado =  s  ∧ (∀ i : Z)(0 ≤ i <  s  → resultado[i] = s[i] + n ) }					
Método de partición por categorías					
1) Problemas a testear		7) Armar los casos de test			
sumarN		Descripción del caso	valor de n	¿s tiene elementos?	Datos de test
2) Elegir una unidad funcional		secuencia vacía	-	No	secuencia vacía
sumarN		n nulo	n=0	Sí	s
3) Identificar factores		n negativo	n<0	Sí	secuencia de restas de los elementos de s con  n
n: Z , s: seq(Z)		n positivo	n>0	Sí	secuencia de sumas de los elementos de s con  n
4) Determinar categorías					
n:Z -Valor					
s:seq(Z) - ¿Tiene elementos?					
5) Determinar elecciones					
s:seq(Z) -¿Tiene elementos?	n: Z valor				
-Sí	n= 0				
-No	n<0				
	n>0				
6) Clasificar elecciones					
s:seq(Z) -¿Tiene elementos?	n: Z valor				
-Sí [Restricción]	n= 0 [Único]				
-No [Único]	n<0 [Restricción]				
	n>0 [Restricción]				

multiplosDeN :: Integer -> [Integer] -> [Integer]  
que dado un número n y una lista xs, devuelve una lista con los  
elementos de xs múltiplos de n

1) multiplosDeN			7)					
2) multiplosDeN			Descripción del caso	valor de n	¿xs tiene elementos?	cantidad de múltiplos de n en xs	Resultado esperado	Datos de test
3) n.Z, xs: seq(Z)			Múltiplos de 0	0	Si	>1	Lista no vacía con los elementos que son 0 de xs	n=0, xs=[0,1,0,2,0]
4)			Lista vacía	-	No		0 Lista vacía	n=1, xs=[]
n.Z	xs:seq(Z)	relación entre n y xs	Hay un solo múltiplo con n positivo	>0	Si		1 Lista con un múltiplo de n	n= 2, xs=[4,5]
Valor	¿Tiene elementos?	cantidad de múltiplos de n en xs	Hay más de un múltiplo con n positivo	>0	Si	>1	Lista con más de 1 elemento múltiplo de n	n=2, xs=[4,6,8,9]
5)			No hay múltiplos, con n positivo	>0	Si		0 Lista vacía	n=5, xs=[1,2,4]
n.Z	xs:seq(Z)	relación entre n y xs	Hay un solo múltiplo con n negativo	<0	Si		1 Lista con un elemento múltiplo de n	n=-3, xs = [-6]
Valor	¿Tiene elementos?	cantidad de múltiplos de n en xs	Hay más de un múltiplo, n negativo	<0	Si	>1	Lista con más de 1 elemento múltiplo de n	n=-2, xs= [2,4,-6,-8]
n=0	Si	0	No hay múltiplos, con n negativo	<0	Si		0 Lista vacía	n=-3, xs = [-8, 7]
n<0	No	1						
n>0		>1						
6)								
n.Z	xs:seq(Z)	relación entre n y xs						
Valor	¿Tiene elementos?	cantidad de múltiplos de n en xs						
n=0 [Único]	Si	0						
n<0	No [Único]	1 [Único]						
n>0		>1						

ordenar :: [Integer] -> [Integer] que ordena los elementos de la lista en forma creciente

1) ordenar			7)					
2) ordenar			Descripción del caso	¿xs tiene elementos?	¿xs está ordenada crecientemente?	Resultado Esperado	Datos de test	
3) xs: seq(Z)			Lista vacía	No	-	Lista vacía	xs=[]	
4)			Lista ordenada	Si	Si	xs	xs=[1,2,3]	
xs:seq(Z)			Lista desordenada	Si	No	Lista de los elementos de xs ordenados crecientemente	xs=[1,5,4,3,6]	
¿Tiene elementos?	¿Está ordenada crecientemente?							
5)								
xs:seq(Z)								
¿Tiene elementos?	¿Está ordenada crecientemente?							
Si	Si							
No	No							
6)								
xs:seq(Z)								
¿Tiene elementos?	¿Está ordenada crecientemente?							
Si	Si [Único]							
No [Único]	No							

aplanarConNBlaños :: [[Char]] -> Integer -> [Char], que a partir de una lista de palabras y un entero n,  
arma una lista de caracteres concatenándolas e insertando n blancos entre cada palabra (n debe ser no negativo)

1) aplanarConNBlaños			7)					
2) aplanarConNBlaños			Descripción del caso	¿xs tiene elementos?	Valor de n	¿N es no negativo?	Resultado esperado	Datos de test
3) xs:seq<seq<Char>>, n.Z			Lista vacía	No	-	-	Lista vacía	xs=[], n=2
4)			Sin espacio en blanco	Si		0 Si	Lista de caracteres de xs concatenados sin espacios	xs=["Ahora","Mañana"],n=0
xs:seq<seq<Char>>, n.Z	n.Z		Tiene espacio en blanco	Si	>0	Si	Lista de caracteres de xs concatenados con n espacios	xs=["Ahora","Mañana"], n=2
¿Tiene elementos?	Valor		N no válido	-	<0	No	Error	n= -2
¿Es no negativo?								
5)								
xs:seq<seq<Char>>, n.Z	n.Z	n.Z						
¿Tiene elementos?	Valor	¿Es no negativo?						
Si	0 Si							
No	>0 No							
	<0							
6)								
xs:seq<seq<Char>>, n.Z	n.Z	n.Z						
¿Tiene elementos?	Valor	¿Es no negativo?						
Si	0 [Único] Si							
No [Único]	>0 No [Error]							
	<0 [Error]							