1. .
   1. El objetivo de realizar testing en un software es el de poder encontrar defectos en el mismo.
   2. Si al realizar testing sobre un software, no se obtienen fallas en los resultados obtenidos, no nos garantiza que funcione correctamente, quizás si para los casos de test concretos, pero no necesariamente.
   3. El testing de caja negra consiste en derivar los casos de test a partir de la especificación del problema, y no su solución.
   4. No se puede realizar testing de caja negra sin contar con una especificación, ya que va contrario a su definición.
2. Dados tres enteros que se interpretan como la longitud de cada uno de los lados de un triángulo, se determina si el triángulo resultante es isósceles, escaleno o equilátero

**Factores**: lado1, lado2, lado3.

**Características**: valor.

**Elecciones**:

* Factor: lado1
  + Característica: Valor.
    - Elección: Es menor o igual a 0 [ERROR].
    - Elección: Es igual a lado2 e igual a lado3 [UNICO].
    - Elección: Es igual a lado2 [RESTRICCION].
    - Elección: Es igual a lado3 [RESTRICCION].
* Factor: lado2
  + Característica: Valor.
    - Elección: Es menor o igual a 0 [ERROR].
    - Elección: Es igual a lado1 e igual a lado3 [UNICO].
    - Elección: Es igual a lado1 [RESTRICCION].
    - Elección: Es igual a lado3 [RESTRICCION].
* Factor: lado3
  + Característica: Valor.
    - Elección: Es menor o igual a 0 [ERROR].
    - Elección: Es igual a lado1 e igual a lado2 [UNICO].
    - Elección: Es igual a lado1 [RESTRICCION].
    - Elección: Es igual a lado2 [RESTRICCION].

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Casos | Lado1  y/o Lado2 y/o Lado3 menores o iguales a 0 | Lado1 = Lado2 = Lado3 | Lado1 = Lado2 | Lado1 = Lado3 | Lado2 = Lado3 | Resultado esperado | Comentarios |
| Caso 1: Lado1= 1; Lado2 = 3, Lado3 = 0 | Si | No | No | No | No | Error. | Para que exista un triangulo en primer lugar, la distancia entre un lado y otro, debe ser mayor estricta que 0. |
| Caso 2: Lado1 = 1, Lado2 = 1, Lado3 = 1 | NO | SI | SI | SI | SI | Equilátero. |  |
| Caso 3: Lado 1, Lado2 = 1, Lado 3 = 2 | NO | NO | SI | NO | NO | Isósceles. |  |
| Caso 4: Lado 1, Lado2 = 2, Lado 3 = 1 | NO | NO | NO | SI | NO | Isósceles. | No es relevante cual sea igual a cuál, solo que dos factores compartan el mismo valor y otro no. |
| Caso 5: Lado 1, Lado2 = 2, Lado 3 = 2 | NO | NO | NO | NO | SI | Isósceles. | No es relevante cual sea igual a cuál, solo que dos factores compartan el mismo valor y otro no. |
| Caso 6: Lado 1, Lado2 = 2, Lado 3 = 3 | NO | NO | NO | NO | NO | Escaleno. | Mientras no sean menores o iguales a 0, y los tres valores sean iguales. |

1. problema parteEntera (x: R): Z {requiere: {True} asegura: {resultado ≤ x < resultado + 1}}

**Factores**: X

**Características**: ¿Es igual a 0? ¿Es menor igual a X?

**Elecciones**:

* Factores: X
  + ¿Es igual a 0?
    - SI [UNICO]
    - NO
  + ¿Es menor igual a X?
    - SI
    - No [ERROR]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Casos | Es igual a 0 | Es menor igual a X | Resultado | Comentarios |
| 0 | SI | SI | 0 | Es correcto. |
| 0.54 | NO | SI | 0 | Es correcto. |
| 1.34 | NO | SI | 1.34 | Error. |

1. problema quitarTodos (e: T, s: seq⟨T⟩): seq⟨T⟩ {requiere: {True} asegura: {resultado es igual a s pero sin el elemento e.}}

**Factores**: e, s.

**Características**: ¿S este vacío? ¿Pertenece e a s?

**Elecciones**:

* ¿S este vacío?
  + SI [UNICO]
  + NO
* ¿Pertenece e a s?
  + SI
  + NO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Casos | S esta vacío | Pertenece e a s | Resultado | Comentarios |
| E = 1, S = [] | SI | NO | [] | Esta bien ya que al estar vacío, E no puede pertenecer a S. |
| E = 2, S = [1,3] | NO | NO | [1,3] | Esta bien ya que 2 no es elemento de S |
| E = 1, S = [1,3] | NO | SI | [3] | Como 1 es elemento de S, el resultado es S sin el elemento 1. |
| E = 2, S = [1,2] | NO | SI | [1,2] | Error, el resultado no debería incluir a 2 como elemento de S. |

1. problema sumarN (n: Z, s: seq⟨Z⟩): seq⟨Z⟩ {requiere: {True} asegura: {|resultado| = |s| ∧ (∀i: Z) (0 ≤ i < |s| → resultado[i] = s[i] + n}}

Factores: n, s.

Características: ¿S este vacío? ¿n es 0?

Elecciones:

* ¿S este vacío?
  + SI [UNICO]
  + NO
* ¿N es 0?
  + NO [RESTRICCION]
  + SI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Casos | S esta vacío | N es 0 | Resultado | Comentarios |
| N = 1, S = [] | SI | NO | [] | Como S esta vacío, N no suma a ningún elemento. |
| N = 0, S = [] | SI | NO | [] | Como S este vacío, N no suma a ningún elemento. |
| N = 1, S = [1,2] | NO | NO | [2,3] | N no es 0 y S no está vacío, así N le suma a cada elemento. |

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

Factores: numero, listaEnteros, relación entre número y listaEnteros.

Características: valor ¿listaEnteros está vacía? ¿Cantidad de múltiplos de numero de listaEnteros?

Elecciones:

* Factor: numero
  + Valor:
    - Menor que 0.
    - Igual a 0 [UNICO].
    - Mayor que 0.
* Factor: listaEnteros
  + ¿Esta vacía?
    - SI [UNICO].
    - NO.
* Factor: Relación entre número y listaEnteros.
  + ¿Cantidad de múltiplos de número y listaEnteros?
    - 0.
    - 1 [UNICO].
    - Mayor a 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Casos | Valor | Esta vacía | Cantidad de múltiplos de número y listaEnteros | Resultado |
| Caso 1: Múltiplos de 0 | =0 | NO | >1 | La lista del resultado es no vacía y contiene solo ceros que tiene la lista original en la misma cantidad. |
| Caso 2: Lista vacía | >0 | SI | 0 | El resultado es una lista vacía. |
| Caso 3: Hay un solo múltiplo. | >0 | NO | 1 | El resultado es una lista de un solo elemento, el cual es el elemento múltiplo con número. |
| Caso 4: No hay múltiplos con numero negativo. | <0 | NO | 0 | El resultado es una lista vacía que no hay múltiplos de número que sean negativos. |
| Caso 5: Hay mas de un múltiplo con numero negativo. | <0 | NO | >1 | El resultado es una lista no vacía con los elementos múltiplos de numero contenidos originalmente por listaEnteros. |
| Caso 6: No hay múltiplos con numero positivo | >0 | NO | 0 | El resultado es una lista vacía ya que no hay múltiplos de numero en listaEnteros. |
| Caso 7: Hay mas de un múltiplo con numero positivo. | >0 | NO | >1 | El resultado es una lista de longitud mayor a 1 y cuyos elementos son múltiplos de numero y están incluidos originalmente en listaEnteros. |