Resolución práctica 1, primera parte

A continuación presentamos soluciones a ejercicios seleccionados de la Práctica 1 - primera parte.

Como siempre, en el proceso de resolución de un problema, primero debemos dedicar un tiempo al análisis del mismo para asegurarnos de entender correctamente qué se pide. Luego, pasamos a la resolución propiamente dicha.

Recomendamos usar este documento como un elemento de ayuda y referencia. Es importante que no lean la solución sin antes haber leído y entendido el problema, y sin haber dedicado un tiempo a pensar cómo resolverlo.

Por otro lado, no hay una única forma correcta de escribir una solución.

1 Proposiciones y Condicionales

Veremos resoluciones a algunos de los ejercicios de la sección 1.

1.1 Ejercicio 3

En este ejercicio debemos clasificar un triángulo, representado por el valor de sus tres ángulos interiores, en equilátero, isósceles o escaleno, según corresponda. Para ello, debemos definir una función, que podemos llamar `clasifica-triángulo`, con tres argumentos numéricos y que devuelva un string: "equilátero", "isósceles" o "escaleno".

Un triángulo es equilátero cuando sus tres lados tienen igual longitud. Si tiene dos lados de igual longitud, decimos que es isósceles; mientras que si los tres lados tienen longitudes distintas, es escaleno.

> Notar que siguiendo la definición, todo triángulo equilátero es isósceles. Sin embargo, cuando un triángulo sea equilátero, daremos como respuesta "equilátero", ya que es la categoría más restrictiva.

Para determinar si un triángulo es equilátero, isósceles o escaleno, en lugar de comparar las longitudes de sus lados, podemos comparar las amplitudes de sus ángulos. Si notamos con `a1`, `a2` y `a3` al valor de los ángulos interiores del triángulo, entonces podemos decir que:

1. es equilátero si todos sus ángulos son iguales, es decir `a1 = a2 = a3` (notar que entonces cada uno sus ángulos interiores debe ser igual a `60°`);

2. es isósceles si dos de sus ángulos son iguales, es decir si `a1 = a2` o `a2 = a3` o `a3 = a1`;

3. es escaleno en cualquier otro caso, es decir cuando todos los ángulos son distintos entre sí.

Teniendo esto en mente, podemos definir nuestra función de la siguiente manera:

```
(define (clasifica-triángulo a1 a2 a3)
        (if (and (= a1 a2) (= a2 a3))
            "equilátero"
            (if (or (= a1 a2) (= a2 a3) (= a3 a1))
                "isósceles"
                "escaleno")))
```

Podemos testear nuestra función con algunos valores:

```
> (clasifica-triángulo 50 80 50)
"isósceles"
> (clasifica-triángulo 80 50 50)
"isósceles"
> (clasifica-triángulo 50 50 80)
"isósceles"
> (clasifica-triángulo 60 60 60)
"equilátero"
> (clasifica-triángulo 50 60 70)
"escaleno"
```

¿Qué resultado obtenemos si evaluamos con los valores `70`, `70` y `70` o `50`, `50` y `90`? ¿Es correcto? Veamos el próximo ejercicio.

1.2 Ejercicio 4

El ejercicio 4 nos pide ampliar la función anterior para verificar que los ángulos proporcionados correspondan a un triángulo, es decir, que **la suma de sus ángulos sea de 180 grados**.

Por lo tanto, primero debemos verificar que efectivamente los ángulos pertenecen a un triángulo. Una vez comprobado esto, podemos utilizar la función definida en el ejercicio anterior (sin modificarla) para la clasificación del triángulo.

Así, la nueva función nos quedaría definida de esta forma:

```
(define (clasifica-triángulo-con-error a1 a2 a3)
        (if (= (+ a1 a2 a3) 180)
            (clasifica-triángulo a1 a2 a3)
            "Error: No es un triángulo"))
```

Notemos que reutilizamos la función del ejercicio 3 para el caso en que sea un triángulo y si no, devolvemos un mensaje de error.

Testeemos nuestra función con algunos valores

```
> (clasifica-triángulo-con-error 40 40 100)
"isósceles"
> (clasifica-triángulo-con-error 60 50 55)
"Error: No es un triángulo"
> (clasifica-triángulo-con-error 60 60 60)
"equilátero"
> (clasifica-triángulo-con-error 80 35 65)
"escaleno"
> (clasifica-triángulo-con-error 80 35 64)
"Error: No es un triángulo"
> (clasifica-triángulo-con-error -50 100 130)
"escaleno"
> (clasifica-triángulo-con-error -10 200 -10)
"isósceles"
```

Prestemos especial atención a los dos últimos casos. ¿Son correctos esos resultados? ¿Es posible que alguno de los ángulos de un triángulo sea negativo? ¿Cómo modificaría la función para tener en cuenta estos casos?

1.3 Ejercicio 7

Recordemos que una terna pitagórica consiste de tres números que podrían ser las longitudes de los lados de un triángulo rectángulo. Esto ocurrirá si satisfacen la ecuación del Teorema de Pitágoras. Es decir, dadas las longitudes `a`, `b` y `c`, tiene que valer alguna de las siguientes igualdades:

$$a^2 = b^2 + c^2$$

$$b^2 = a^2 + c^2$$

$$c^2 = a^2 + b^2$$

Una opción es probar sucesivamente si alguna de las tres expresiones es verdadera. Si una lo es, entonces tenemos una terna pitagórica. Si no, probamos con la siguiente igualdad. Si las tres son falsas, los valores no forman una terna pitagórica. De esta manera, la siguiente sería una posible solución:

```
(define (pitagorica? a b c) (if (= (sqr a) (+ (sqr b) (sqr c)))
                                #true
                                (if (= (sqr b) (+ (sqr a) (sqr c)))
                                    #true
                                    (if (= (sqr c) (+ (sqr a) (sqr b)))
                                        #true
                                        #false))))
```

Veamos qué pasa si probamos la función con algunos ejemplos:

```
> (pitagorica? 3 4 5)
#true
> (pitagorica? 10 6 8)
#true
> (pitagorica? 12 15 9)
#true
> (pitagorica? 1 2 3)
#false
> (pitagorica? 1 1 1)
#false
```

Otra alternativa es pensar que en una terna pitagórica la longitud más grande es la que sería la longitud de la hipotenusa. De esta forma, si encontramos cuál es la mayor de las tres longitudes, solamente debemos revisar la ecuación que la eleva al cuadrado y suma los cuadrados de las otros dos. Una solución que use este enfoque podría ser:

```
(define (pitagorica? a b c) (if (and (> a b) (> a c))
                                (= (sqr a) (+ (sqr b) (sqr c)))
                                (if (> b c)
                                    (= (sqr b) (+ (sqr a) (sqr c)))
                                    (= (sqr c) (+ (sqr a) (sqr b))))))
```

En esa función, primero preguntamos si `a` es mayor que `b` y `c`. En caso de que lo sea, entonces sabremos si los números son una terna pitagórica si consideramos la fórmula en la que `a` es la hipotenusa. En caso contrario, sabremos que al menos uno entre `b` y `c` es mayor o igual a `a`. Entonces, vemos cuál es el máximo entre ellos y revisamos la fórmula que lo considere como la hipotenusa.

Comprobar que esta definición da los mismos resultados que la anterior. Probar los ejemplos ya vistos y algunos nuevos.