Programación Declarativa

Ingeniería Informática
Especialidad de Computación
Cuarto curso. Primer cuatrimestre

Escuela Politécnica Superior de Córdoba
Universidad de Córdoba

Curso académico: 2015 – 2016

Práctica número 2: Predicados y sentencias condicionales

1. Codifica un predicado denominado **pertenece-intervalo?** que compruebe si un número real **x** pertenece al intervalo delimitado por otros dos números **a** y **b**.
   - (**pertenece-intervalo?** a b x) devuelve **#t** si y sólo si "$a \ x \leq b$"
     - *Por ejemplo:* (**pertenece-intervalo?** 1 9 2) ➔ #t

2. Codifica un predicado denominado ***propiedad-triángular?*** que compruebe si tres lados pueden construir un triángulo al cumplir la propiedad triangular. Se ha de tener en cuenta que, en un triángulo, **cualquier lado** es menor que la suma de los otros dos y mayor que el valor absoluto de su diferencia, es decir, se ha de verificar **cualquiera** de las siguientes desigualdades:
     - $|a - b| < c < a + b$
     - $|a - c| < b < a + c$
     - $|b - c| < a < b + c$

   - *Ejemplos:*
     - *(**forma-triángulo?** 3 4 5)* ➔ *#t*
     - *(**forma-triángulo?** 1 2 19)* ➔ *#f*

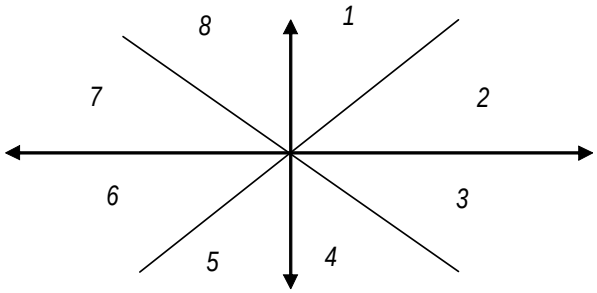
3. **(*)** Codifica una función denominada **redondear** de forma que si "n" es un número entero y "d" es su parte decimal entonces se debe verificar que
   - si ($0.0 \leq d < 0.5$)
     - entonces *(redondear n.d)* ➔ *n*
     - en caso contrario *(redondear n.d)* ➔ *n+1*

4. (*) Utiliza la forma especial ***case*** para definir una función que permita calcula **la letra del DNI**.
     - La función recibirá como parámetro el número
     - y deberá devolver la letra que le corresponde.

5. **(*)** Codifica un predicado denominado **bisiesto**? que reciba como parámetro a un número y determine si corresponde o no a un año bisiesto, teniendo en cuenta que:
   - Un año es bisiesto si es divisible por 4 pero no es divisible por 100:
     (**bisiesto?** 2008) => #t
   - Un año es bisiesto si es divisible por 100 y además es divisible por 400:
     (**bisiesto?** 1600) => #t
   - Nota: por tanto, los años que son divisibles por 100 pero no son divisibles por 400 no son bisiestos.
     - Por ejemplo: (**bisiesto?** 1900) => #f

6. Codifica la función **octante** para que indique en qué octante se encuentra ubicado un punto

P(x,y) del plano:

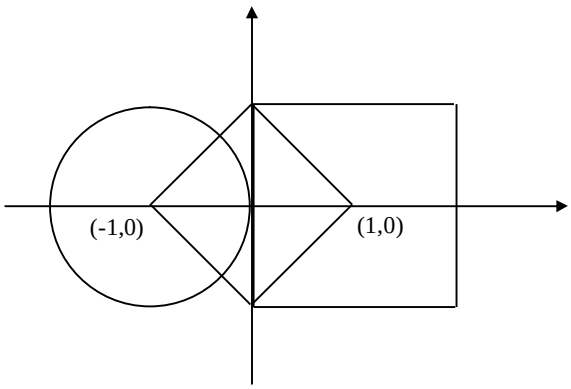


Si el punto pertenece al eje de coordenadas o a algunas de las bisectrices de los cuadrantes entonces (*octante x y) tomará el valor cero*

7. (*) Predicado ***corona-circular***?:
   - Considérense dos circunferencias concéntricas con centro el punto (3,3) y radios 1 y 3, respectivamente.
   - La función **corona-circular** debe devolver el valor
     - **#t** si el punto P(x,y) está fuera de la circunferencia de radio 1 y dentro de la circunferencia de radio 3;
     - en cualquier otro caso, el valor devuelto debe ser **#f.**

8. (*) Se dispone de dos circunferencias concéntricas cuyo centro es el punto (0, 2) y con radios 1 y 3, respectivamente. Codifica la siguiente función *f*:
   - Los puntos interiores a la circunferencia de radio 1 tendrán el valor
     $$f(x,y) = (x - y)/2$$
   - Los puntos interiores a la circunferencia de radio 3 y exteriores a la circunferencia de radio 1 tendrán el valor
     $$f(x,y) = (x + y)/2$$
   - Los puntos exteriores a la circunferencia de radio 3 tendrán el valor
     $$f(x,y) = x + y$$
   - Los puntos que pertenezcan a alguna de las dos circunferencias tendrán el valor $f(x,y) = 1$

9. **(*)** Dada las siguientes figuras geométricas



   - Codifica una función que asigne a un punto P(x,y) el valor que le corresponde según su posición:
     0. el punto pertenece a uno de los lados del rombo o del cuadrado o a la

circunferencia.

1. el punto pertenece solamente al círculo
2. el punto pertenece al círculo y al rombo.
3. el punto pertenece al rombo, pero no pertenece al círculo ni al cuadrado.
4. el punto pertenece al rombo y al cuadrado.
5. el punto pertenece solamente al cuadrado.
6. en otro caso.

- **Observación**: se deben utilizar las funciones auxiliares que calculan las distancias **euclídea**, de **Manhattan** y de **ajedrez**.

10. Indica los valores que resultan al aplicar los predicados primitivos.

```
(boolean? #t)      (boolean? #f)   (boolean? (> 2 3))      (boolean? (+ 2 4))
(number? 3)        (define a 2)    (number? a)
(negative? a)      (positive? a)   (zero? a)
(even? a)          (even? (+ a 1)) (odd? a)        (odd? (+ a 1))

(define (par? x)
        (= 0 (remainder x 2))
)
(procedure? par?)  (procedure? 'par?)

(complex? 3+4i)    (complex? 3)
(real? 3.5)        (real? 3.2+0.0i)    (real? 3+4i)
(rational? 6/10)   (rational? 3)       (rational? 3+4i)
(integer? 2)       (integer? 3.2)      (integer? 3/5)
(define letra1 "w") (define letra2 'w) (define letra3 #\w)
(char? letra1)     (string? letra1)    (char? letra2)
(string? letra2)   (char? letra3)      (string? letra3)
(char? "w")        (string? "w")       (char? #\w)
(string? #\w)
```

11. Comprueba los resultados de los siguientes predicados de equivalencia:

```
(eq? 9/2 9/2)      (eqv? 9/2 9/2)      (equal? 9/2 9/2)

(define a 9/2)     (define b 9/2)
(eq? a b)          (eqv? a b)          (equal? a b)    (= a b)

(define a 3)       (define b 3)
(eq? a b)          (eqv? a b)          (equal? a b)    (= a b)

(define a 3)       (define b 3.)       (define c 3.0)
(eq? a b)          (eq? a c)           (eq? b c)
(eqv? a b)         (eqv? a c)          (eqv? b c)
(equal? a b)       (equal? a c)        (equal? b c)
(= a b)            (= a c)             (= b c)

(define a (+ 3. 2)) (define b (+ 3 2.))
(eq? a b)          (eqv? a b)          (equal? a b))   (= a b)

(define a "dato")  (define b "dato")
(eq? a b)          (eqv? a b)          (equal? a b))

(define a (cons 'a 'b)) (define b (cons 'a 'b))
(eq? a b)          (eqv? a b)          (equal? a b)

(define a (lambda (x) (* x x)))        (define b (lambda (x) (* x x)))
(eq? a b)          (eqv? a b)          (equal? a b)
```

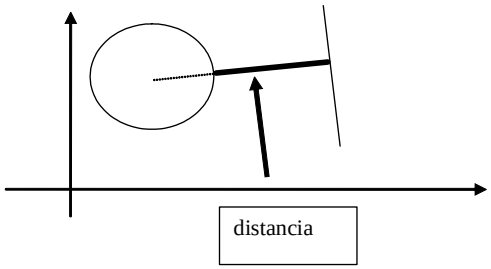
12. **(*)** Codifica un predicado denominado **alineados?** que reciba como parámetros las coordenadas de tres puntos del plano y compruebe si están alineados.
    - Ejemplo:
        - (**alineados?** 0 0 1 1 7 7) ➔ #t
        - (**alineados?** 0 0 1 4 -9 2) ➔ #f

13. **(*)** Codifica una función que calcule el área de un triángulo según la fórmula de **Herón**.
    - La función recibirá como parámetros las coordenadas de los vértices.
    - La función deberá comprobar **previamente** si los puntos están o no alineados.
        - Si los puntos están alineados, el área, obviamente, será cero;
        - en caso contrario, se les aplicará la fórmula de Herón.

14. **(*)** Codifica una función denominada **posición-circunferencia-recta** que determine la posición relativa de una circunferencia y una recta:
        - Exterior: 1
        - Secante: 2
        - Tangente: 3

    - La función recibirá como parámetros:
        - El radio y las coordenadas del centro de la circunferencia.
        - Los coeficientes de la recta r: ax + b y + c = 0
    - Sugerencia: utiliza la función auxiliar que calcula la distancia de un punto a una recta.

15. **(*)** Codifica una función, denominada **distancia-circunferencia-recta**, que calcule la distancia entre una circunferencia y una recta.



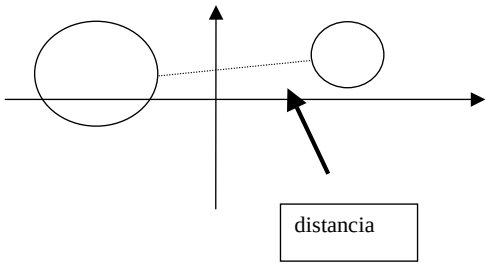
    - La función recibirá como parámetros
        - El radio y las coordenadas del centro de la circunferencia.
        - Los coeficientes de la recta r ≡ ax + b y + c = 0
        - Notas:
            1. Utiliza la función auxiliar que calcula la distancia de un punto a una recta.
            2. Si la recta es secante a la circunferencia entonces la distancia debe ser cero.

16. **(*)** Codifica una función que determine la posición relativa de dos círculos.
    - La función recibirá como parámetros las coordenadas de los centros y los radios devolverá los siguientes valores:
        - Iguales: 0
        - Secantes: 1
        - Tangentes por dentro: 2
        - Tangentes por fuera: 3
        - Interiores: 4
        - Exteriores: 5
        - Concéntricas: 6

17. Codifica una función, denominada **distancia**, que calcule la distancia entres dos

circunferencias.

- La función recibirá como parámetros el radio y las coordenadas del centro de cada circunferencia.



- Nota: si las circunferencias no son exteriores, la distancia debe ser cero.

18. **(*)** Codifica un predicado denominado **lados-paralelos?** que reciba como parámetros las coordenadas de cuatro puntos y determine si la recta determinada por los dos primeros puntos es paralela a la recta que determinan los otros dos puntos.

19. **(*)** Codifica una función que calcule el **área de un trapecio**:
    - La función ha de recibir como parámetros las coordenadas de los vértices.
    - La función deberá determinar **previamente** qué lados forman las bases utilizando el predicado **lados-paralelos?**
    - Observación
        - La función deberá utilizar la función que calcula la distancia de un punto a una recta para poder calcular la altura del trapecio.

20. **(*)** Codifica un predicado denominado **perpendiculares?** que reciba cuatro puntos y que determine si la recta que pasa por los dos primeros es perpendicular a la que pasa por los dos últimos.

21. **(*)** Utiliza la forma especial ***let*** para codificar una función que calcule el área de un rombo:
    - La función ha de recibir como parámetros las coordenadas de los vértices del rombo.
    - La función deberá usar el predicado **perpendiculares?** para determinar previamente qué vértices forman las diagonales del rombo.