Examen de Diseño de Algoritmos

Se valorará tanto la organización del código, como la claridad de exposición.

Considere el problema de representar un objeto que sea un triangulo en un plano. Cada triangulo puede ser representado por tres puntos, cada uno con dos coordenadas.

1) *(2 puntos) Defina un constructor llamado crea-triangulo y selectores llamados punto1, punto2 y punto3. Cada punto tiene que estar definido con cons. Programe también un procedimiento llamado perimetro, de forma que devuelva la suma de las longitudes de los lados*

Recordemos que la notación '.' se utiliza para indicar que una función puede tener varios argumentos. Por ejemplo:

```
(define (f x y . z)
  (begin
    (print x)
    (print y)
    (print z)
    ))

(f 1 2 3 4) ; <- Devuelve 1'(3 4)
```

2) *(2 puntos) Defina un procedimiento llamado suma-iguales que tome uno o más números enteros y que sume los números que tengan la misma paridad que el primero. De una versión recursiva y otra iterativa.*

Un bucle do-while se puede ver como un procedimiento que toma tres argumentos: un procedimiento auxiliar, una lista de valores y un test. En el do-while se aplica el procedimiento al primer elemento de la lista y se testa si el elemento de la lista cumple el test para realizar otro bucle.

El siguiente ejemplo es una visión de como funcionaria en Scheme el do-while:

```
(do-while
 (lambda (x) (newline) (display x))
 (list 57 321 88 10 80)
 (lambda (x) (> x 30)
 )

57
321
88
10
```

3) *(3 puntos) Programe una implementación de do-while. El valor retornado no es importante.*

Lea el siguiente link y luego intente la siguiente pregunta.

En vez de implementar una cola como un par de punteros, se puede hacer una pila como un procedimiento con variables locales. Las variables locales consisten en los punteros al principio y al fin de la cola. Por lo que una posible implementación sería de la siguiente manera:

```
(define (make-queue)
  (let ((front-ptr ... )
        (rear-ptr ... ))
    < procedimientos internos >
    (define (dispatch m) ...)
    dispatch))
```

4) *(3 puntos) Complete el código de forma que se implementen todas las operaciones de la cola en esta representación.*