Tarea 1
CC4101 - Lenguajes de Programación

Éric Tanter
Auxiliar: Richard Ibarra

Fecha de Entrega. Viernes 16 de Abril

Diseñando abstracciones

1. **(1 punto)** Defina la función `filter`, la cual filtra elementos en una lista en base a un predicado que recibe como parámetro. No se olvide especificar su contrato/firma de manera precisa, así como empezar por implementar un conjunto de tests[1].

```
> (filter odd? '(1 2 3 4 5 6))
(1 3 5)
```

2. **(1 punto)** Una desventaja de su definición de `filter` es que siempre hay que volver a especificar el predicado. Mejore la definición de `filter` separando los dos parámetros, usando la tecnica conocida como *currying*[2] (*currificación*[3] en español). No se olvide especificar la nueva firma y actualizar/extender los tests.

```
> (define greater-than-5 (filter (lambda (x) (> x 5))))
> (greater-than-5 '(2 4 6 8 10))
(6 8 10)
> (greater-than-5 '(1 2 3 4))
()
```

3. **(0.5 puntos)** Ahora queremos definir una función `lookup`, la cual retorna el primer elemento en una lista que cumple un predicado (con currificación). Para esto, podemos reusar nuestra abstracción de filtrado para definir `lookup`. Hágalo. ¿Cuál es la desventaja de usar `filter` para definir `lookup`?

[1] Ver el documento "como diseñar programas" entregado al inicio del semestre.
[2] http://en.wikipedia.org/wiki/Currying
[3] http://es.wikipedia.org/wiki/Currificación

4. **(1 punto)** `filter` demostró no ser la abstracción adecuada para definir `lookup`. Entonces, podemos subir el nivel de abstracción un paso más, de manera de abstraer el qué hacer cuando se cumple el predicado. Defina la función `process`, especificando con cuidado su contrato/firma y unos tests de uso.

5. **(1 punto)** Ya que `process` abstrae `filter` y `lookup`, redefina ambas funciones usando `process`. Explique por qué esa nueva definición de `lookup` es adecuada (en comparación con la definición del punto 3.).

6. **(0.5 puntos)** Use la función `process` para definir la función `member` que determina si un elemento se encuentra en una lista.

Reglas entrega

- Recuerde que su tarea es individual y será penalizada con nota 1 los trabajos que sean copias.
- La entrega se debe hacer vía U-Cursos hasta las 23.59 del Viernes 16 de Abril.
- Una función no comentada **se ignorará por completo** (se borrará el código entregado)

Presentación (1 pto)

Usted deberá presentar su tarea frente al Auxiliar o Ayudante del curso en donde demuestre que domina los puntos que pretende controlar esta Tarea. En esta presentación deberá responder las preguntas que se hacen a lo largo del enunciado. Se anunciará oportunamente los horarios en que podrá realizar su presentación.

Reglas presentaciones

- La presentación de la tarea **es obligatoria** y no debe durar más de 5 minutos.
- Deberá usar la última versión entregada en U-Cursos. No **se aceptará mostrar otra versión.**
- La presentación puede causar que su nota disminuya o se mantenga, por tanto, es su responsabilidad prepararse tanto como estime conveniente.