

## EIF400 – Paradigmas de Programación

### Proyecto de programación #1

Profesor: Mag. Georges Alfaro S.

#### PLANTEAMIENTO DEL PROBLEMA

---

Para este proyecto deberán implementar un conjunto de funciones que permitan realizar varias operaciones con polinomios.

#### OBJETIVOS DEL PROYECTO

---

El proyecto tiene como objetivo estudiar algunas de las técnicas fundamentales de programación funcional, en especial técnicas de uso de parámetros y recursividad.

#### DESCRIPCIÓN DEL PROYECTO

---

##### Descripción general.

Un polinomio de grado  $n$  puede representarse por medio de una lista simple.

El polinomio:

$$p_n(x) = \sum_{i=0}^n a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

se representa usando la lista de coeficientes:

$$(a_0 \ a_1 \ \dots \ a_n)$$

Para un polinomio de grado  $n$ , la lista contiene  $n + 1$  elementos. Aunque la lista puede contener coeficientes que valen 0, ésta no puede contener ceros a la derecha, ya que representaría un polinomio de grado menor. **Cuando una lista tenga ceros a la derecha, debe simplificarse adecuadamente.**

Por ejemplo, la lista:

$$(0 \ 2 \ 3 \ 1)$$

es una representación del polinomio:

$$2x + 3x^2 + x^3$$

La lista:

$$(1 \ 0 \ -1 \ 0 \ 0)$$

corresponde al polinomio:

$$1 - x^2$$

y debe simplificarse como:

$$(1 \ 0 \ -1)$$

### Operaciones básicas.

Se construirán las siguientes funciones:

Función	Descripción
<b>(display-p p)</b>	<b>Despliegue de polinomios.</b> La función mostrará el polinomio utilizando una notación estándar. Por ejemplo, la invocación: $(display-p \ '(1 \ 3 \ 0 \ -2))$ producirá como resultado: $1 + 3x + -2x^3$ Observen que el exponente no se muestra cuando vale 1 y el factor constante, (que corresponde al exponente 0) no despliega la variable x.
<b>(+p p1 p2 ...)</b>	<b>Suma de polinomios.</b> La función recibe una cantidad arbitraria de polinomios en forma de una lista de coeficientes, los suma y regresa el polinomio resultante. Si el polinomio resultante es de menor grado que alguno de los utilizados como argumento de la función, éste debe simplificarse correctamente.
<b>(-p p1 p2 ...)</b>	<b>Resta de polinomios.</b> La función recibe una cantidad arbitraria de polinomios en forma de una lista de coeficientes y regresa el polinomio resultante. <b>La semántica de la función corresponde con la función aritmética de resta.</b> Ya que la operación no es asociativa, los parámetros se asociarán desde la izquierda, como en la función resta. Es decir, la operación: $(-p \ p1 \ p2 \ p3)$ Debe interpretarse correctamente como: $(- \ p \ (-p \ p1 \ p2) \ p3)$

(*p p1 p2 ...))	<b>Multiplicación de polinomios.</b> La función recibe una cantidad arbitraria de polinomios en forma de una lista de coeficientes, calcula el producto y regresa el polinomio resultante.
(qt-p p1 p2)	<b>División de polinomios (parte 1).</b> Esta función calcula el <b>cociente</b> de la división de los polinomios recibidos como parámetro. Observe que esta función recibe <b>solamente 2 parámetros</b> .
(rem-p p1 p2)	<b>División de polinomios (parte 2).</b> Esta función calcula el <b>residuo</b> de la división de los polinomios recibidos como parámetro. Observe que esta función recibe <b>solamente 2 parámetros</b> .
(/-p p1 p2)	<b>División de polinomios (parte 3).</b> Esta función calcula el resultado de la división de los polinomios recibidos como parámetro. El resultado es una lista que contiene el cociente y el residuo de la división.
(drv-p p1 p2 ...)	<b>Derivación de polinomios.</b> La función recibe una cantidad arbitraria de polinomios en forma de una lista de coeficientes, <b>deriva</b> cada polinomio y regresa una lista con el resultado.
(eval-p p x)	<b>Evaluación de polinomios.</b> La función evalúa el polinomio $p$ en el valor $x$ . La función <b>no utilizará ninguna función para elevar números a ninguna potencia</b> , ya sea una función de biblioteca o definida en el programa. Para evaluar el polinomio, se utilizarán <b>multiplicaciones repetidas</b> . Este método se conoce como el <b>algoritmo de Horner</b> . <a href="https://es.wikipedia.org/wiki/Algoritmo_de_Horner">https://es.wikipedia.org/wiki/Algoritmo_de_Horner</a>

(fact-p p1)	<p><b>Factorización de polinomios.</b></p> <p>Esta función recibe un <b>único polinomio</b> como parámetro y produce una lista con <b>todos los polinomios que sean factores</b> del primero.</p> <p>Aunque un polinomio real puede tener raíces conjugadas (raíces imaginarias), <b>la función sólo considerará raíces reales.</b></p> <p>La función de multiplicación (*p) debe poder aplicarse a la lista resultante para obtener el polinomio original.</p> <p>Por ejemplo:</p> <pre>(fact-p '(0 2 0 -2))</pre> <p>producirá como resultado:</p> <pre>'((1 -1) (1 1) (0 2))</pre> <p>Este resultado representa:</p> $(1 - x)(1 + x)(2x) = 2x - 2x^3$ <p>El polinomio:</p> $(2 + x)^2 = 4 + 4x + x^2$ <p>se escribe:</p> <pre>'(4 4 1)</pre> <p>y se factoriza como:</p> <pre>'((2 1) (2 1))</pre> <p>Observe que, aunque ambos monomios son el mismo, aparecen repetidos en el resultado.</p> <p>Los polinomios de grado 2 y 3 pueden factorizarse directamente, utilizando la fórmula general correspondiente.</p> <p><a href="https://es.wikipedia.org/wiki/Ecuaci%C3%B3n_de_segundo_grado">https://es.wikipedia.org/wiki/Ecuaci%C3%B3n_de_segundo_grado</a></p> <p><a href="https://es.wikipedia.org/wiki/Ecuaci%C3%B3n_de_tercer_grado">https://es.wikipedia.org/wiki/Ecuaci%C3%B3n_de_tercer_grado</a></p>
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## CONSIDERACIONES DE IMPLEMENTACIÓN

---

**Escriba el programa utilizando el lenguaje de programación DrRacket<sup>1</sup>.**

Incluya en el código un bloque de información sobre el proyecto y las personas que presentan la solución, de la siguiente manera:

```
; -----  
;  
;  
; (c) 2024  
; EIF400 Paradigmas de Programación  
; 2do ciclo 2024  
; Proyecto #1  
;  
; (cédula y nombre de la persona)  
; (cédula y nombre de la persona)  
; (cédula y nombre de la persona)  
;  
;  
; version 1.0.0 2024-10-19  
;  
; -----
```

Agregue además los comentarios necesarios para explicar la implementación de cada función definida, e incluya las pruebas que considere necesarias para verificar su correcto funcionamiento.

No puede utilizar sentencias de control estructuradas ni variables locales al implementar cada función. Cualquier valor intermedio que deba evaluar, será pasado como parámetro a una función. Tampoco es permitido emplear arreglos. Deberá usar solamente listas como estructura básica de datos para la representación de los polinomios.

Es posible utilizar funciones propias del lenguaje que no se hayan visto en clase, pero no se podrán usar bibliotecas o funciones que implementen de alguna manera las solicitadas.

**Todos los polinomios deben mostrarse siempre simplificados.**

Todas las operaciones que no pueden resolverse completamente, como, por ejemplo:

$$\frac{2 - x}{1 + 3x^2}$$

---

<sup>1</sup> Recuerde indicar la opción '#racket' en el código fuente.

deben quedar indicadas, y producir un resultado evaluable, por ejemplo:

`(rem-p '(2 -1) '(1 0 3))`

---

## ENTREGA Y EVALUACIÓN

---

El proyecto debe entregarse **por medio del aula virtual, en el espacio asignado para ello**. La entrega se hará al finalizar la semana 13 del curso. (**domingo 20 de octubre de 2024**). No se aceptará ningún proyecto después de esa fecha, ni se admitirá la entrega del proyecto por correo electrónico. El proyecto se puede realizar en grupos de **cuatro personas, como máximo**.

En caso de que las funciones no trabajen adecuadamente, efectúe un análisis de los resultados obtenidos, indicando las razones por las cuales el programa no se ejecuta correctamente, y cuáles son las posibles correcciones que se podrían hacer.

El proyecto se evaluará de acuerdo con la siguiente ponderación:

Funcionalidad	Puntos	Porcent.
Despliegue de polinomios	2	2,0%
Suma y resta de polinomios	6	6,0%
Suma de polinomios Resta de polinomios	4	4,0%
	2	2,0%
Multiplicación de polinomios	8	8,0%
División de polinomios	20	20,0%
Derivación de polinomios	8	8,0%
Evaluación de polinomios	10	10,0%
Factorización de polinomios	46	46,0%
Polinomios de grado 2 Polinomios de grado 3 Polinomios de grado superior	8	8,0%
	12	12,0%
	26	26,0%
	100	100,0%

Observaciones generales:

- Se debe incluir en **cada documento entregado** el nombre completo y cédula de cada participante del grupo, indicando el nombre del curso, ciclo lectivo y descripción del trabajo que se entrega, de la manera indicada.
- Cualquier trabajo práctico que no sea de elaboración original de los estudiantes (plagio) se calificará con nota 0 (cero) y se procederá como lo indiquen los reglamentos vigentes de la universidad.
- Los trabajos que se reciban después de la fecha señalada para su entrega, **en caso de ser aceptados, serán penalizados con un 30% de la nota por cada día de atraso.**