Evidencias y Ejercicios

"A10 y A11"

Departamento de Ciencias de la Computación

Asignatura:

"Lenguajes Inteligentes"

Profesor:

Alejandro Padilla Díaz

Fecha:

4 de octubre de 2024

Alumnos:

Juan Francisco Gallo Ramírez

ID: 23287

Ingeniería en Computación Inteligente

5to Semestre

Evidencias

```racket
#lang racket
;========================================= EJERCICIO #1 =====
;-------------------------------------------------------------
;*************** a) Raíz cuadrada de un número:
(sqrt 4)
(sqrt 2)
(sqrt -1)
;*************** b) Computar el Seno de un ángulo:
(sin 15)
(sin 90)
(sin 120)
;*************** c) Determinar el máximo de dos números:
(max 4 5)
(max 4 9 (max 4 (+ 8 2) 9 10 15) 8 10 1)
(+ 9 10 (- 8 2))
(* 9 2 (+ 4 5) (* 5 2) 3)
(/ (+ 6 8) (* 2 2))
(* 2 (- 4 1) (/ 6 18))

;=============================================================
```

```
Welcome to DrRacket, version 8.14 [cs].
Language: racket, with debugging; memory limit: 128 MB.
2
1.4142135623730951
0+1i
0.6502878401571168
0.8939966636005579
0.5806111842123143
5
15
25
4860
3 1/2
2
>
```

```racket
#lang racket
;========================================= EJERCICIO #2 =====
;-------------------------------------------------------------
; Convertir una cantidad de dólar a euros.
;-------------------------------------------------------------

;*************** Función para convertir:
(define (Dolar-Euro cantidad_dolar)
  (/ (* cantidad_dolar 20) 25))

;*************** Ejemplos de conversión:
(Dolar-Euro 10)
(Dolar-Euro 5)
(Dolar-Euro 20)
(Dolar-Euro 22)
(Dolar-Euro 35)

;=============================================================
```

```
Welcome to DrRacket, version 8.14 [cs].
Language: racket, with debugging; memory limit: 128 MB.
8
4
16
17 3/5
28
>
```

```racket
#lang racket
;========================================== EJERCICIO #3 =====
;------------------------------------------------------------
; Elaborar una función recursiva para calcular factorial.
;------------------------------------------------------------

;*************** Función factorial.
(define factorial
  (lambda (n)
    (if (zero? n) 1
        (* n (factorial (sub1 n))))))

;*************** Ejemplos de factorial:
(factorial 20)
(factorial 0)
(factorial 100)

;============================================================
```

```
Welcome to DrRacket, version 8.14 [cs].
Language: racket, with debugging; memory limit: 128 MB.
2432902008176640000
1
933262154439441526816992388562667004907159682643816214685929638952175999932299156089414639761565182862536979208272237582511852109168640000000000000000000
>
```

```racket
#lang racket
;========================================== EJERCICIO #5 =====
;------------------------------------------------------------
; Definir el programa der suma-monedas. Consume cuatro
; números: el número de monedas de $1, $2, $5 y $10 en la
; bolsa. Produce la cantidad de dinero en la bolsa.
;------------------------------------------------------------

;*************** Definición de la función.
(define (suma_monedas m1 m2 m3 m4)
  (+ (* m1 1) (* m2 2) (* m3 5) (* m4 10) ))

;*************** Ejemplos de la función:
(suma_monedas 100 40 30 20)
(suma_monedas 23 40 200 10)

;============================================================
```

```
Welcome to DrRacket, version 8.14 [cs].
Language: racket, with debugging; memory limit: 128 MB.
530
1203
>
```

```
#lang racket
;======================================== EJERCICIO #4 =====
;-----------------------------------------------------------
;*************** Definición de la función.
(define (convert3 num1 num2 num3)
  (+ (* num3 100) (* num2 10) num1))
;*************** Ejemplos de la función:
(convert3 1 1 1)
(convert3 5 3 4)
;*************** a) Evaluar n^2 + 10:
(define (g n)
  (+ (* n n) 10))
; Ejemplos:
(g 2)
(g 5)
(g 9)
;*************** b) Evaluar (1/2)*n^2 + 20:
(define (h n)
  (+ (* (/ 1 2) (* n n)) 20))
; Ejemplos:
(h 2)
(h 5)
(h 9)
;*************** c) Evaluar 2 - (1/n):
(define (b n)
  (- 2 (/ 1 n)))
; Ejemplos:
(b 2)
(b 5)
(b 9)
;===========================================================
```

```
Welcome to DrRacket, version 8.14 [cs].
Language: racket, with debugging; memory limit: 128 MB.
111
435
14
35
91
22
32 1/2
60 1/2
1 1/2
1 4/5
1 8/9
>
```