Disciplina: **AARE Paradigmas de Linguagens de Programação**
Professor: Prof. Ausberto S. Castro V.
E-mail: ascv@uenf.br

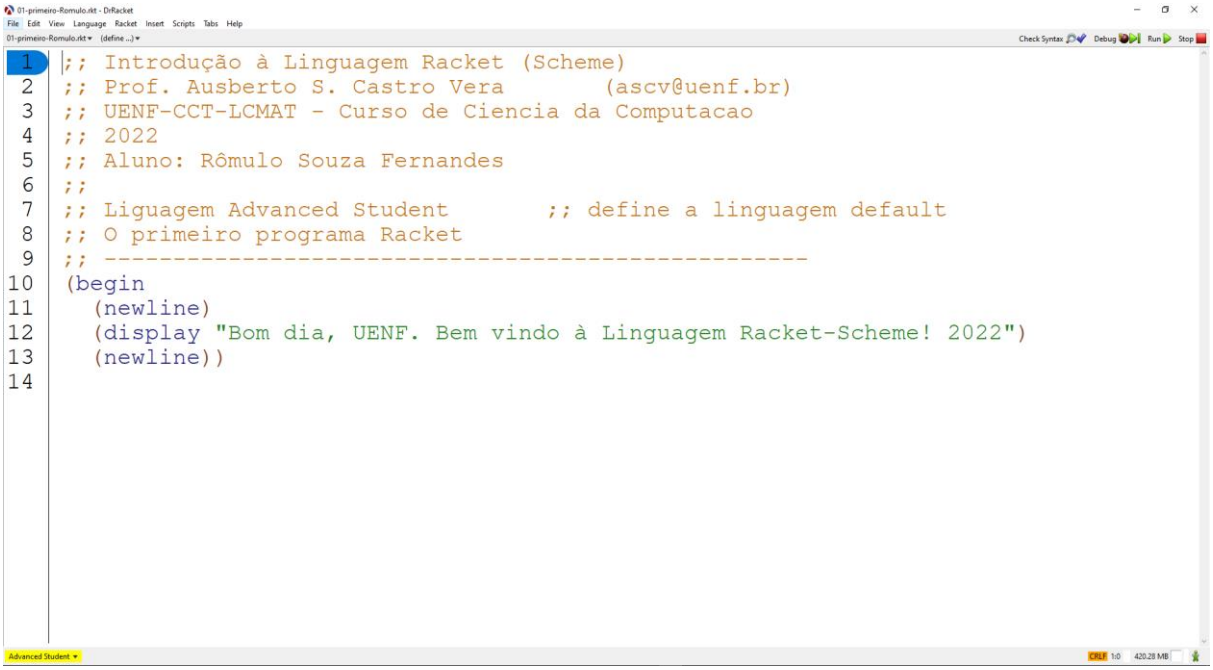
Data: 23 de setembro de 2022

Prática – Racket

Nome Completo: Rômulo Souza Fernandes
Data: 23 de setembro de 2022 Total Exercícios Resolvidos: .....

Arquivo 01-primeiro-Romulo.rtk Primeiro programa em Racket

1. Execute o programa e indicar o que faz cada linha do código fonte do programa. Quais funções estão definidas nas 4 linhas. Explique cada uma delas?



```
;; Introdução à Linguagem Racket (Scheme)
;; Prof. Ausberto S. Castro Vera        (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2022
;; Aluno: Rômulo Souza Fernandes
;;
;; Liguagem Advanced Student        ;; define a linguagem default
;; O primeiro programa Racket
;; ---------------------------------------------------
(begin
  (newline)
  (display "Bom dia, UENF. Bem vindo à Linguagem Racket-Scheme! 2022")
  (newline))
```

Codigo fonte

```
;; 2022
;; Aluno: Rômulo Souza Fernandes
;;
;; Liguagem Advanced Student       ;; define a linguagem default
;; O primeiro programa Racket
;; ------------------------------------------------
(begin
  (newline)
  (display "Bom dia, UENF. Bem vindo à Linguagem Racket-Scheme! 2022")
  (newline))
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student; memory limit: 128 MB.

Bom dia, UENF. Bemvindo à Linguagem Racket-Scheme! 2022
>
```

Interações (shell)

Explicação

begin: O begin é usado para indicar o inicio do programa.

newline: O newline tem a função de quebrar linhas.

display: O display tem a função de printar na tela o texto que estiver entre aspas.

2. Agregar linhas de código para mostrar na parte executável, a mensagem “Pratica 01 – Linguagem Racket” , o nome completo do aluno e a data atual

```
;; Introdução à Linguagem Racket (Scheme)
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2022
;; Aluno: Rômulo Souza Fernandes
;;
;; Liguagem Advanced Student       ;; define a linguagem default
;; O primeiro programa Racket
;; ------------------------------------------------
(begin
  (newline)
  (display "Bom dia, UENF. Bem vindo à Linguagem Racket-Scheme! 2022")
  (newline)
  (display "Pratica 01 - Linguagem Racket")
  (newline)
  (display "Nome: Rômulo Souza Fernandes")
  (newline)
  (display "20 de Setembro de 2022")
  (newline))
```

```
;; Introdução à Linguagem Racket (Scheme)
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2022
;; Aluno: Rômulo Souza Fernandes
;;
;; Liguagem Advanced Student       ;; define a linguagem default
;; O primeiro programa Racket
;; ------------------------------------------------
(begin
  (newline)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student; memory limit: 128 MB.

Bom dia, UENF. Bem vindo à Linguagem Racket-Scheme! 2022
Pratica 01 - Linguagem Racket
Nome: Rômulo Souza Fernandes
20 de Setembro de 2022
>
```

Arquivo 02-numeros.rtk Números e Aritmética

3. Execute o programa e mostre os resultados

```
;; Introdução à Linguagem Racket (Scheme)
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2022
;; Aluno: Rômulo Souza Fernandes
;;
;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;

; ----------------------------------------------
(display "    UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "    Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "    Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
;;
;; Numeros e aritmetica
;; ------------------------------------------------

(display "Soma 23 + 28 = ")
(+ 23 28)

(display "Produto 14*17 = ")
(* 14 17)

(display "Combinando 5 + (3*7) = ")
(+ 5 (* 3 7))

(display "Combinando (2 + (3*4))/2 - 4 = ")
(- (/ (+ 2 (* 3 4)) 2) 4)
```

```
(display "Raiz quadrada de 4= ")
(sqrt 4)

(display "Raiz quadrada de 2= ")
(sqrt 2)

(display "Complexos - raiz quadrada de -1= ")
(sqrt -1)

;; #i significa "inexato"
(newline)
(display "Valor de Pi+1 ")
(+ pi 1)

(display "Seno 90 graus: ")
(sin (/ pi 2))

(display "Coseno 60 graus: ")
(cos (/ pi 3))

(display "Coseno 45 graus: ")
(cos (/ pi 4))

(display "Logaritmo Natural de 15: ")
(log 15)

(display "exponente 2^3 = ")
(expt 2 3)

(display "exponente 4^(1/2) = ")
(expt 4 1/2)
```

```
(display "exponente 2^3 = ")
(expt 2 3)

(display "exponente 4^(1/2) = ")
(expt 4 1/2)

(display "Maximo de  1 3 4 2 3 = ")
(max 1 3 4 2 3)

(display "minimo de  1 3 4 2 3 = ")
(min 1 3 4 2 3)

(display "valor absoluto de 3 = ")
(abs  3)

(display "valor absoluto de -4 = ")
(abs -4)

;; Expressoes "quote": listas de objetos tratados como dados
;;
(newline)
'"quotes obriga as listas serem tratadas como DADOS"
(quote ( 2 4 6))
(quote (/ 4 (* 3 7)))
'(1 2 3 4)    ;; comentario
'((a b)(3 5))
'(+ 2 (* 5 7) )
```

```
(abs -4)

;; Expressoes "quote": listas de objetos tratados como dados
;;
(newline)
'"quotes obriga as listas serem tratadas como DADOS"
(quote ( 2 4 6))
(quote (/ 4 (* 3 7)))
'(1 2 3 4)    ;; comentario
'((a b)(3 5))
'(+ 2 (* 5 7) )
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student; memory limit: 128 MB.
    UENF-CCT-LCMAT-CC, 2022
    Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
    Aluno:  Rômulo Souza Fernandes
Soma 23 + 28 = 51
Produto 14*17 = 238
Combinando 5 + (3*7) = 26
Combinando (2 + (3*4))/2 - 4 = 3
Raiz quadrada de 4= 2
Raiz quadrada de 2= #i1.4142135623730951
Complexos - raiz quadrada de -1= 0+1i

Valor de Pi+1 #i4.141592653589793
Seno 90 graus: #i1.0
Coseno 60 graus: #i0.5000000000000001
```

```
(abs -4)

;; Expressoes "quote": listas de objetos tratados como dados
;;
(newline)
'"quotes obriga as listas serem tratadas como DADOS"
(quote ( 2 4 6))
(quote (/ 4 (* 3 7)))
'(1 2 3 4)    ;; comentario
'((a b)(3 5))
'(+ 2 (* 5 7) )
```

```
Coseno 60 graus: #i0.5000000000000001
Coseno 45 graus: #i0.7071067811865476
Logaritmo Natural de 15: #i2.70805020110221
exponente 2^3 = 8
exponente 4^(1/2) = 2
Maximo de  1 3 4 2 3 = 4
minimo de  1 3 4 2 3 = 1
valor absoluto de 3 = 3
valor absoluto de -4 = 4

"quotes obriga as listas serem tratadas como DADOS"
(list 2 4 6)
(list '/ 4 (list '* 3 7))
(list 1 2 3 4)
(list (list 'a 'b) (list 3 5))
(list '+ 2 (list '* 5 7))
```

4. Escreva programas Racket para as seguintes expressões:

4.1. H = $(4 - (7^2 + 6^3) / 3) - (6 + (5 - (2^4 - 8))$

```
;; Introdução à Linguagem Racket (Scheme)
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2022
;; Aluno: Rômulo Souza Fernandes
;;
;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;
; ----------------------------------------------
(display "UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)
(display "H = (4 - (7^2 + 6^3) / 3) - (6 + (5 - (2^4 - 8)))")
(newline)
(display "Resultado:")
(newline)
(display "H = ")
(- (- 4 (/(+(expt 7 2) (expt 6 3))3)) (+ 6 (- 5(- (expt 2 4) 8))))
```

```
;;
; ----------------------------------------------
(display "UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)
(display "H = (4 - (7^2 + 6^3) / 3) - (6 + (5 - (2^4 - 8)))")
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student; memory limit: 128 MB.
UENF-CCT-LCMAT-CC, 2022
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
Aluno:  Rômulo Souza Fernandes

H = (4 - (7^2 + 6^3) / 3) - (6 + (5 - (2^4 - 8)))
Resultado:
H = -87.3
>
```

4.2. Escreva um NOVO programa Racket que calcule o valor da expressão:

$$\frac{\sqrt{7^2+5+\sin(18-7)+\cos(20+2)}}{(5+3)*(4-8)^2}$$

```
;; Introdução à Linguagem Racket (Scheme)
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2022
;; Aluno: Rômulo Souza Fernandes
;;
;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;
; ------------------------------------------------
(display "UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "Aluno: Rômulo Souza Fernandes ")
(newline)
(display (/ (sqrt (+ (+ (+ 49 5)(sin (- 18 7)))(cos (+ 20 2)))) (*(+ 5 3))16 ))
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student; memory limit: 128 MB.
UENF-CCT-LCMAT-CC, 2022
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
Aluno: Rômulo Souza Fernandes

0.05633676520454323
>
```

Arquivo 03-variaveis.rtk

5. Execute o programa e indique o valor das variáveis m, z, k

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes      <===========  escreva seu nome aqui

;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno: Rômulo Souza Fernandes")
(newline)
;;
;;
;; Variáveis e Expressoes LET
;; ------------------------------------------------
(define x 3)                        ; para x=3
(define m 7)
(define z (+ m 6))                  ; para z = m+ 6
(define Pi 3.141516)
(define k (+ z (- m x)))

(define (quadrado x)            ; (define   (nomeFuncao   parametro)  definicao  )
    (* x x)  )


(define (entre5e20? n)
  (and (< 5 n) (< n 20) )               ;; AND logico
 )
```

```
Welcome to DrRacket, version 8.6 [cs].
```

```
;; ---------------------------------------------------

(newline)
(display "O numero 8 esta entre 5 e 20? ")
(entre5e20? 8)

(newline)
(display "O numero 37 esta entre 5 e 20? ")
(entre5e20? 37)

(newline)
(display "Escreva qualquer variavel definida acima ... <var> <ENTER>   x m z Pi")

;; ---------------------------------------------------
; Expressao let:    (let ((var1 valor) ...) expr1  expr2 .....)
(newline)
(let ((x 24))
  (+ x 6))

;;;
(let ( (a 5) (b 8))
  (+ 3 (* a b)))

;;;
(let ( (op1 +)
       (op2 *)
       (x 5)
     )
  (op1 3 (op2 4 x))
)

(display "O quadrado de 12 : ")(quadrado 12)
```

```
Welcome to DrRacket, version 8.6 [cs].
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes      <===========  escreva seu nome aqui

;;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;
#lang racket      ;; define a linguagem default
; ---------------------------------------------
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno: Rômulo Souza Fernandes

O numero 8 esta entre 5 e 20? #t

O numero 37 esta entre 5 e 20? #f

Escreva qualquer variavel definida acima ... <var> <ENTER>   x m z Pi
30
43
23
O quadrado de 12 : 144
> m
7
> z
13
> k
17
>
```

m = 7
z = 13
k = 17

5.1. Explicar o significado de cada uma das 3 expressões de iteração `let`

R = A expressões de iteração `let` são usadas em linguagens com paradigma funcional, com a função de restringir variáveis locais em uma subexpressão.

5.2. Escreva 2 expressões do tipo `let` e explique o seu significado

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;Escolha a linguagem "Determine language from source"
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno: Rômulo Souza Fernandes")
(newline)
;;
(define a 0)
(define b 0)

(let ((a 2))
  (* a 4))

(let ((b 15))
  (- a b))
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;Escolha a linguagem "Determine language from source"
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno: Rômulo Souza Fernandes")
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno: Rômulo Souza Fernandes
8
-15
>
```

Arquivo 04-areas.rtk

6. Execute o programa e indique o que faz o programa

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------

(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
(newline)
;;
;;
;; (define   (nomeFuncao   parametros)  definicao  )
;;
;; AREAS
;; ------------------------------------------------
(define (area-circulo r)
  (* 3.14 (* r r))         )

(define (areatriangulo b h)
  (/ (* b h) 2))

(define (areadisco interno externo)
  (- (area-circulo externo)
     (area-circulo interno)
```

Welcome to DrRacket, version 8.6 [cs].

```
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
(newline)
;;
;;
;; (define   (nomeFuncao   parametros)  definicao  )
;;
;; AREAS
;; ------------------------------------------------
(define (area-circulo r)
  (* 3.14 (* r r))         )

(define (areatriangulo b h)
  (/ (* b h) 2))

(define (areadisco interno externo)
  (- (area-circulo externo)
     (area-circulo interno)
  )
)
;
;-------------------------------------------------------------

(display "Circulo de raio 18   AREA = ")
(area-circulo 18)

(display "Triangulo base=10, altura=15  AREA =   ")
(areatriangulo 10 15)

(display "Disco raio menor=12, raio maior=20  AREA = ")
(areadisco 12 20)
```

Welcome to DrRacket, version 8.6 [cs].

```
(define (areadisco interno externo)
  (- (area-circulo externo)
     (area-circulo interno)
  )
)
;
;-------------------------------------------------------------

(display "Circulo de raio 18   AREA = ")
(area-circulo 18)

(display "Triangulo base=10, altura=15  AREA =   ")
(areatriangulo 10 15)

(display "Disco raio menor=12, raio maior=20  AREA = ")
(areadisco 12 20)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

Circulo de raio 18   AREA = 1017.36
Triangulo base=10, altura=15  AREA =   75
Disco raio menor=12, raio maior=20  AREA = 803.8399999999999
>
```

O programa acima calcula a área interna de formas geométricas e a área interna entre circulos.

6.1. Escreva um programa Racket para calcular a área de um quadrado qualquer, a área de um trapézio e a área de um polígono.

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------

(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
(newline)
(define (area-quadrado r )
  (* r r))
(display "Área de um quadrado de lado 5: ")
        (area-quadrado 5)

(newline)
(define (area-trapezio base_maior base_menor h )
  (/(*(+ base_maior base_menor) h ) 2))
```

```
UENF-CCT-LCMAT-CC, 2022
```

```
; ----------------------------------------------

(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
(newline)
(define (area-quadrado r )
  (* r r))
(display "Área de um quadrado de lado 5: ")
        (area-quadrado 5)

(newline)
(define (area-trapezio base_maior base_menor h )
  (/(*(+ base_maior base_menor) h ) 2))

(display "Área de um trapezio de base maior=5, base menor=3 e altura=1: ")
(area-trapezio 5 3 1)

(define (area-poligono p a )
  (* (* 0.5 p) a))
(newline)
(display "Área de um poligono de perimetro=50 e apótema=5: ")
(area-poligono 50 5)
```

```
; ----------------------------------------------

(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
(newline)
(define (area-quadrado r )
  (* r r))
(display "Área de um quadrado de lado 5: ")
        (area-quadrado 5)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

Área de um quadrado de lado 5: 25

Área de um trapezio de base maior=5, base menor=3 e altura=1: 4

Área de um poligono de perimetro=50 e apótema=5: 125.0
>
```

6.2. Escreva um programa NOVO completo para calcular o volume de um galão de óleo utilizando a fórmula $V = \pi R^2 A$, onde as variáveis V, R e A representam, respectivamente, o volume, o raio e a altura.

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------

(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
(newline)

(define (Vgalao R A )
  (*(*(* R R)A)3.14))
(display "O volume de um galão com raio = 8 e altura = 19: ")
(newline)
(Vgalao 8 19)
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------

(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

O volume de um galão com raio = 8 e altura = 19:
3818.2400000000002
>
```

Arquivo 05-funcoes.rtk

7. Execute o programa e explique o que faz

```racket
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;;;;;;;;;;;;;;;   Escolha a linguagem R5RS
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
;;
;; DEFINICAO DE FUNCOES-procedimentos  usando lambda
;;       (define  NomeFuncao  (lambda  ( parametros )  (definicao) )  )
;;-------------------------------------------------

(define ADICIONA
      (lambda  (a  b)
        (+ a b )
       )
)
;;
;;
(define soma
  (lambda (x y)
    (begin
      (newline)
```

```racket
      (display x)
       (display " e ")
       (display y)
       (display " = ")
      (+ x y)
     )
   )
)
;;-------------------------------------------
(define produto
  (lambda (x y)
    (begin
      (newline)
      (display "O produto de ") (display x) (display " e ") (display y)
      (display " = ")
      (* x y)
     )
   )
)
;;-------------------------------------------
(define divisao
  (lambda (x y)
    (begin
      (newline)
      (display "A divisao ") (display x) (display "/") (display y)(display " = ")
      (/ x y)
     )
   )
)
;;-------------------------------------------
(define diferenca
  (lambda (x y)
    (begin
```

```
61        )
62      )
63  )
64  ;;---------------------------------------------
65  (define diferenca
66    (lambda (x y)
67      (begin
68        (newline)
69        (display "A diferenca ") (display x) (display "-") (display y)(display " = ")
70        (- x y)
71       )
72     )
73  )
74  ;;---------------------------------------------
75  (define quadrado
76      (lambda ( a )
77        (newline)
78       (display "O quadrado de ") (display a) (display " = ")
79        ( * a a)
80        )
81    )
82  ;;---------------------------------------------
83
84  ;;----------- Executando funcoes -------------
85
86  (ADICIONA 25 15)
87  (ADICIONA 30 40)
88
89  (soma 45 32)
90  (produto 21 15)
91  (divisao 420 7)
92  (diferenca 89 35)
93  (quadrado 09)
```

```
61        )
62      )
63  )
64  ;;---------------------------------------------
65  (define diferenca
66    (lambda (x y)
67      (begin
68        (newline)
69        (display "A diferenca ") (display x) (display "-") (display y)(display " = ")
70        (- x y)
71       )
72     )
73  )
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes
40
70

A soma de 45 e 32 = 77

O produto de 21 e 15 = 315

A divisao 420/7 = 60

A diferenca 89-35 = 54

O quadrado de 9 = 81
>
```

O programa 05-funções utiliza funções anônimas pela palavra reservada lambda, assim realizando operações aritméticas como soma, produto, divisão, diferença e quadrado.

7.1. Escreva um programa onde é definido duas funções

```
  ;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;;;;;;;;;;;;;;;   Escolha a linguagem R5RS
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)

(define produto
  (lambda (x y)
    (begin
      (newline)
      (display "Produto de ") (display x) (display " e ") (display y)
      (display " = ")
      (* x y)
     )
   )
)
(produto 2 4)

Produto de 2 e 4 = 8
```

```
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)

(define produto
  (lambda (x y)
    (begin
      (newline)
      (display "Produto de ") (display x) (display " e ") (display y)
      (display " = ")
      (* x y)
     )
   )
)
(produto 2 4)

(define divisao
  (lambda (x y)
    (begin
      (newline)
      (display "Divisao de ") (display x) (display "/") (display y)(display " = ")
      (/ x y)
     )
   )
)
(divisao 2 4)

Produto de 2 e 4 = 8
```

```
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)

(define produto
  (lambda (x y)
    (begin
      (newline)
      (display "Produto de ") (display x) (display " e ") (display y)
      (display " = ")
      (* x y)
    )
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes


Produto de 2 e 4 = 8

Divisao de 2/4 = 1/2
>
```

Arquivo 06-condicionalIF.rtk

8. Executar e explicar o programa

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;;;;;;;;;;;;;;;   Escolha a linguagem R5RS
;;
;; Ajuda:  http://docs.racket-lang.org/guide/syntax-overview.html#(part._.Conditionals_with_if__and__or__and_cond)
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
;;
;;
;; Condicionais    (if  (condicao) (alternativa1) (alternativa2 ) )
;; ---------------------------------------------------

(define P 126)
(display "P = ")
P
(if (> P 57)
    (display "P maior que 57")
    (display "P menor que 57")
    )
;;---------------------------

(define (reply s)
  (if (equal? "Oi" (substring s 0 2))
      "Tudo Bem?"
```

```
    )
;;----------------------------

(define (reply s)
  (if (equal? "Oi" (substring s 0 2))
      "Tudo Bem?"
      "Nao entendi."))

;;----------------------------
(define quadrado
     (lambda ( a )
       ( * a a)
       )
  )
;;----------------------------
(newline)
(define minquadrado
             (lambda (a b)
             (if (< a b)
                        (quadrado a)
                        (quadrado b) ) ) )

;;----------------------------

(display "MinQadrado de 6 e 9 = ") (minquadrado 6 9)
(newline)

(display "Oi pessoal da Computação UENF! ")
(reply "Oi pessoal da Computação UENF!")

(display "Vai chover na madrugada? ")
(reply "Vai chover na madrugada?")
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;;;;;;;;;;;;;;;   Escolha a linguagem R5RS
;;
;; Ajuda:  http://docs.racket-lang.org/guide/syntax-overview.html#(part._.Conditionals_with_if__and__or__and_cond)
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
;;
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes
P = 126
P maior que 57
MinQadrado de 6 e 9 = 36

Oi pessoal da Computação UENF! "Tudo Bem?"
Vai chover na madrugada? "Nao entendi."
>
```

O programa 06-condicionalIF realiza comparações de valores e strings utilizando o condicional IF, funciona da seguinte forma, é definido um comando a ser realizado a uma condição, tendo duas alternativas possíveis para o valor ou string que for comparado, true e false, de acordo com o resultado ele realiza o comando do true ou false.

8.1. Escreva um programa com dois condicionais

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;;;;;;;;;;;;;;;   Escolha a linguagem R5RS
;;
;; Ajuda:  http://docs.racket-lang.org/guide/syntax-overview.html#(part._.Conditionals_with_if__and__or__and_cond)
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
;;
;;
;; Condicionais    (if  (condicao) (alternativa1) (alternativa2 ) )
;; ---------------------------------------------------
(newline)

(define media 80)
(display "A média para aprovação é = 60")
(newline)
```

Welcome to DrRacket, version 8.6 [cs].

```
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
;;
;;
;; Condicionais    (if  (condicao) (alternativa1) (alternativa2 ) )
;; ---------------------------------------------------
(newline)

(define media 80)
(display "A média para aprovação é = 60")
(newline)
(if (>= media 60)
    (display "Aluno aprovado!")
    (display "Aluno reprovado!")
    )

(newline)
(newline)

(define num -5)
(if (< num 0)
    (display "O número é negativo!")
    (display "O número é positivo!")
    )
```

Welcome to DrRacket, version 8.6 [cs].

```
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
;;
;;
;; Condicionais    (if  (condicao) (alternativa1) (alternativa2 ) )
;; ---------------------------------------------------
(newline)

(define media 80)
(display "A média para aprovação é = 60")
(newline)
(if (>= media 60)
    (display "Aluno aprovado!")
    (display "Aluno reprovado!")
    )
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

A média para aprovação é = 60
Aluno aprovado!

O número é negativo!
>
```

8.2. Escreva um programa para calcular a média de três notas e indique “Aprovado” se for maior ou igual a 6,0, e “Reprovado”, caso contrário

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes

;;;;;;;;;;;;;;;;   Escolha a linguagem R5RS
;;
;; Ajuda:  http://docs.racket-lang.org/guide/syntax-overview.html#(part._.Conditionals_with_if__and__or__and_cond)
;;
#lang racket      ;; define a linguagem default
; ---------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes")
(newline)
;;
;;
;; Condicionais    (if  (condicao) (alternativa1) (alternativa2 ) )
;; ---------------------------------------------------
(newline)

(define nota1 8)
(define nota2 6)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
```

```
16 (newline)
17 (display "  Aluno:  Rômulo Souza Fernandes")
18 (newline)
19 ;;
20 ;;
21 ;; Condicionais    (if  (condicao) (alternativa1) (alternativa2 ) )
22 ;; ---------------------------------------------------
23 (newline)
24
25 (define nota1 8)
26 (define nota2 6)
27 (define nota3 5)
28 (define media 0)
29
30 (set! media (/ (+(+ nota1 nota2)nota3)3))
31
32 (display "A média para aprovação é = 6")
33
34 (newline)
35
36 (if (>= media 6)
37     (display "Aluno aprovado!")
38     (display "Aluno reprovado!")
39     )
40
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
```

```
15 (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
16 (newline)
17 (display "  Aluno:  Rômulo Souza Fernandes")
18 (newline)
19 ;;
20 ;;
21 ;; Condicionais    (if  (condicao) (alternativa1) (alternativa2 ) )
22 ;; ---------------------------------------------------
23 (newline)
24
25 (define nota1 8)
26 (define nota2 6)
27 (define nota3 5)
28 (define media 0)
29
30 (set! media (/ (+(+ nota1 nota2)nota3)3))
31
32 (display "A média para aprovação é = 6")
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

A média para aprovação é = 6
Aluno aprovado!
>
```

Arquivo 07-formulas.rtk

9. Executar e explicar o programa

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernande ")
(newline)
;;
;;Algumas formulas conhecidas


;;-----------------------------------
;; Formula de Pitagoras
(define (pitagoras a b)
  (+ (* a a) (* b b))
)

;;-----------------------------------
;; Aproximacao do numero Pi
(define (aprox-pi N)
  (if (zero? N)
      4
      (+ (* 4 (/ (expt -1 N) (+ (* 2 N) 1)))
         (aprox-pi (- N 1)))
```

Welcome to DrRacket, version 8.6 [cs].

```
  )
)

;;---------------------------------------
;; factorial de um numero inteiro N
(define (factorial N)
  (if (zero? N)
      1
      (* N (factorial (- N 1)))
  )
)


(newline)
(display "Pitagoras de 3 e 4 = ")
(pitagoras 3 4)

(newline)
(display "Aproximando Pi com 4 = ")
(aprox-pi 4)

(newline)
(display "Aproximando Pi com 10 = ")
(aprox-pi 10)

(newline)
(display "Factorial de 5 = ")
(factorial 5)
(newline)
(display " Fazer outro teste sobre Pitagoras e outro sobre Fatorial")
(newline)
```

Welcome to DrRacket, version 8.6 [cs].

```
  )
)

;;----------------------------------------
;; factorial de um numero inteiro N
(define (factorial N)
  (if (zero? N)
      1
      (* N (factorial (- N 1)))
  )
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

Pitagoras de 3 e 4 = 25

Aproximando Pi com 4 = 3 107/315

Aproximando Pi com 10 = 3 3380087/14549535

Factorial de 5 = 120

 Fazer outro teste sobre Pitagoras e outro sobre Fatorial
>
```

O programa 07-formular utiliza funções para resolver cálculos matemáticos, como Teorema de Pitágoras, aproximação de Pi e fatorial.

9.1. Escreva um programa que calcule o fatorial de um número de uma forma diferente da apresentada.

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)

(define (fatorial n)
    (if (= n 0)
        1
        (* n (fatorial (- n 1)))))
(fatorial 6)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes
720
>
```

Arquivo 08-condicional.rtk

10. Execute o programa e indique o que faz

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Abril 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
;;
;; CONDICIONAL   ( cond {[ ‹expr-test› ‹expr-eval›* ]}* )
;;---------------------------------------------

(define (taxa quantidade)
    (cond
            ((<= quantidade 1000) 0.040)
            ((<= quantidade 5000) 0.045)
            ((<= quantidade 8000) 0.050)
            (else 1)
            )
  )

;;-----------------------------------------

(newline)
(display "Taxa para R$4.000,00 = ")
(taxa 4000)

(newline)
(display "Taxa para R$6.350,00 = ")
(taxa 6350)

(newline)
(display "Taxa para R$15.500,00 = ")
(taxa 15500)
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Abril 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
;;
;; CONDICIONAL   ( cond {[ ‹expr-test› ‹expr-eval›* ]}* )
;;---------------------------------------------
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

Taxa para R$4.000,00 = 0.045

Taxa para R$6.350,00 = 0.05

Taxa para R$15.500,00 = 1
>
```

O programa 08-condicional como o próprio nome já informa, utiliza condicionais, permitindo a execução de diferentes comandos seguindo as condições definidas anteriormente, o programa demonstra o funcionamento com 3 condições e 3 situações para passar pela condicional.

10.1. Escreva um programa condicional com pelo menos 5 opções

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Abril 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)
(display "Porcentagem de alimentação diária adequada:")
(newline)
(define (Alimentacao refeicao)
     (cond
              ((equal? refeicao "Desejum") "20% da alimentação diária")
              ((equal? refeicao "Lanche da manhã") "5% da alimentação diária")
              ((equal? refeicao "Almoço") "30% da alimentação diária")
              ((equal? refeicao "Lanche da tarde") "15% da alimentação diária")
              ((equal? refeicao "Jantar") "25% da alimentação diária")
              ((equal? refeicao "Ceia") "5% da alimentação diária")
              (else "Opção inválida")
              )
  )

(Alimentacao "Almoço")
```

```
Welcome to DrRacket, version 8.6 [cs].
```

```
(newline)
(display "Porcentagem de alimentação diária adequada:")
(newline)
(define (Alimentacao refeicao)
     (cond
              ((equal? refeicao "Desejum") "20% da alimentação diária")
              ((equal? refeicao "Lanche da manhã") "5% da alimentação diária")
              ((equal? refeicao "Almoço") "30% da alimentação diária")
              ((equal? refeicao "Lanche da tarde") "15% da alimentação diária")
              ((equal? refeicao "Jantar") "25% da alimentação diária")
              ((equal? refeicao "Ceia") "5% da alimentação diária")
              (else "Opção inválida")
              )
  )
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

Porcentagem de alimentação diária adequada:
"30% da alimentação diária"
>
```

10.2. Escreva um programa `bhaskara.rkt` que calcule as raízes de uma equação $25x^2 - 55x+10 = 0$, utilizando a fórmula de Bhaskara. Sugestão: Primeiro faça o algoritmo completo

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Abril 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)
(define (Bhaskara a b c)
 (local ((define delta (- (* b b) (* 4 a c))))
 (cond
 [(< delta 0) empty]
 [else
 (local ((define x1 (/ (- (- b) (sqrt delta)) (* 2 a)))
 (define x2 (/ (+ (- b) (sqrt delta)) (* 2 a))))
 (cond
 [(= delta 0) (list x1)]
 [else (list x1 x2)]))])))
(Bhaskara 1 12 -13)
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Abril 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

'(-13 1)
>
```

Arquivo 09-pares.rtk

11. Execute o programa

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2022")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
;;
;; PARES CONSTANTES
;; ------------------------------------------------

(define x (cons 1 2))
(define y (cons 'a 3))
(define z (cons x y))    ;   (  (1 2) ('a 3) )

(define a (cons 1 2))
(define b (cons 2 1))

(define a2 (cons a 3))
(define b2 (cons 3 b))

(define a3 (cons a2 4))
(define b3 (cons 4 b2))

(define a4 (cons a3 5))
```

```
(define a4 (cons a3 5))
(define b4 (cons 5 b3))
;; ------------------------------------------------
(newline)
(display "O par x = ")
x

(newline)
(display "O par y = ")
y

(newline)
(display "O par z = (x y) = ")
z



(newline)
(display "O par a4 = ")
a4

(newline)
(display "O par b4 = ")
b4
;; ---------------Primeiro e resto --------------------
(newline)
(display "O primeiro elemento do par x = ")
(car x)                ;; primeiro

(newline)
(display "O segundo elemento do par x = ")
(cdr x)                ;; resto
```

```
(define a4 (cons a3 5))
(define b4 (cons 5 b3))
;; ------------------------------------------------
(newline)
(display "O par x = ")
x

(newline)
(display "O par y = ")
y
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2022
  Aluno:  Rômulo Souza Fernandes

O par x = '(1 . 2)

O par y = '(a . 3)

O par z = (x y) = '((1 . 2) a . 3)

O par a4 = '((((1 . 2) . 3) . 4) . 5)

O par b4 = '(5 4 3 2 . 1)

O primeiro elemento do par x = 1

O segundo elemento do par x = 2
>
```

11.1. Escreva um NOVO programa para construir dois pares e indicar em cada um deles o primeiro e o segundo elemento

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2022")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)

(define A (cons 1 2))
(define B (cons 3 4))
(newline)
(display "O par A = ")
A

(newline)
(display "O par B = ")
B
(newline)
(display "Primeiro elemento do par A = ")
(car A)
(newline)
(display "Primeiro elemento do par B = ")
(car B)
```

```
Welcome to DrRacket, version 8.6 [cs].
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2022")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)

(define A (cons 1 2))
(define B (cons 3 4))
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2022
  Aluno:  Rômulo Souza Fernandes


O par A = '(1 . 2)

O par B = '(3 . 4)

Primeiro elemento do par A = 1

Primeiro elemento do par B = 3
>
```

Arquivo 10-listas.rtk e 11-listas.rtk

12. Executar os programas e observe quantos métodos existem para construir listas

10-listas.rtk

```
;; Introdução à Linguagem Racket (Scheme)
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
;; define a linguagem default: R5RS
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
;;
;; LISTAS - Parte 1
;; ----------------------------------------------


(define x (cons 5 6))
(define y (cons 8 9))

;; Utilizando pares de constantes e recursividade

(define lista1 (cons 1 x))

(define lista2 (cons 2 lista1))

(define lista3 (cons 2 lista2))
```

```
(define A (list 1 2 3 4 5))    ; metodo direto
(define B (list 6 7 8 9))      ; metodo direto
(define C (list A B))
(define D (append A B))

;;-------------------------------------
(newline)
(display "Lista A = ")
A

(newline)
(display "Lista B = ")
B

(newline)
(display "Lista C = (list A B)= ")
C

(newline)
(display "Lista D = (append A B)= ")
D

(newline)
(display "Lista D reversa = ")
(reverse D)      ;; lista inversa

(newline)
(display "Comprimento da lista B = ")
(length B)       ;; comprimento de uma lista

;;------------ primeiro elemento y Resto da lista ----------------
(newline)
(display "PRIMEIRO elemento da lista B = ")
(car B)

(newline)
(display "RESTO da lista B = ")
(cdr B)
```

```
(define A (list 1 2 3 4 5))    ; metodo direto
(define B (list 6 7 8 9))      ; metodo direto
(define C (list A B))
(define D (append A B))

;;-------------------------------------
(newline)
(display "Lista A = ")
A
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro
  Aluno:  Rômulo Souza Fernandes

Lista A = '(1 2 3 4 5)

Lista B = '(6 7 8 9)

Lista C = (list A B)= '((1 2 3 4 5) (6 7 8 9))

Lista D = (append A B)= '(1 2 3 4 5 6 7 8 9)

Lista D reversa = '(9 8 7 6 5 4 3 2 1)

Comprimento da lista B = 4

PRIMEIRO elemento da lista B = 6

RESTO da lista B = '(7 8 9)
>
```

11-listas.rtk

```racket
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;

; O procedimento CONS constroi listas e tem DOIS parametros
(cons 'a '(x y w u))

(define lista1 (cons 'a '(x y w u)))

(define lista2a (cons 'x (cons 'a (cons 'm '(3 7)))))
(define lista2b (cons 'x (cons 'a (cons 'm (cons 3 7)))))

(display "Lista2a :") lista2a
(newline)

(display "Lista2b:") lista2b
(newline)

;-------------------------------------------------------
(display "O ponto, antes do ultimo elemento,  numa lista, significa LISTA IMPROPRIA...!")
(newline)
```

Welcome to DrRacket, version 8.6 [cs].

```racket
; O procedimento CONS constroi listas e tem DOIS parametros
(cons 'a '(x y w u))

(define lista1 (cons 'a '(x y w u)))

(define lista2a (cons 'x (cons 'a (cons 'm '(3 7)))))
(define lista2b (cons 'x (cons 'a (cons 'm (cons 3 7)))))

(display "Lista2a :") lista2a
(newline)

(display "Lista2b:") lista2b
(newline)

;-------------------------------------------------------
(display "O ponto, antes do ultimo elemento,  numa lista, significa LISTA IMPROPRIA...!")
(newline)
(cons 'a 'b)

(cdr '(a . b))        ; lista UM elelemento

(cons 'a '(b . c))    ; lista 2 elementos

;-------------------------------------------------------
(newline)
(list 'a 'b 'c)       ; lista 3 elementos

(list 'x)             ; lista unitaria

(list)                ; lista vazia
```

Welcome to DrRacket, version 8.6 [cs].

```
; O procedimento CONS constroi listas e tem DOIS parametros
(cons 'a '(x y w u))

(define lista1 (cons 'a '(x y w u)))

(define lista2a (cons 'x (cons 'a (cons 'm '(3 7)))))
(define lista2b (cons 'x (cons 'a (cons 'm (cons 3 7)))))

(display "Lista2a :") lista2a
(newline)

(display "Lista2b:") lista2b
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes
'(a x y w u)
Lista2a :'(x a m 3 7)

Lista2b:'(x a m 3 . 7)

O ponto, antes do ultimo elemento,  numa lista, significa LISTA IMPROPRIA...!
'(a . b)
'b
'(a b . c)

'(a b c)
'(x)
'()
>
```

12.1. Escreva um NOVO programa para construir uma lista e determinar seu primeiro e último elemento, seu comprimento, e uma nova lista com dois elementos a mais que a anterior. Incluir os códigos fonte

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)

(define lista1 (list 1 2 3 4))
(newline)
(display "Lista 1 = ")
lista1
(newline)
(display "Primeiro elemento da Lista 1 = ")
(car lista1)

(newline)
(display "Último elemento da Lista 1 = ")
(car(cdr (cdr (cdr lista1))))

(newline)
(display "Comprimento da lista 1 = ")
(length lista1)

(define lista2 (list '1 '2 '3 '4 (list '5 '6)))
(newline)
(display "Lista 2 = ")
lista2
```

```
Welcome to DrRacket, version 8.6 [cs].
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)

(define lista1 (list 1 2 3 4))
(newline)
(display "Lista 1 = ")
lista1
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

Lista 1 = '(1 2 3 4)

Primeiro elemento da Lista 1 = 1

Último elemento da Lista 1 = 4

Comprimento da lista 1 = 4

Lista 2 = '(1 2 3 4 (5 6))
>
```

12.2. Utilizando uma ÚNICA linha de comandos, escreva um NOVO programa Racket para construir a lista (4 7 2 9 8 7 1 6 2 3 4) a partir das listas A=(1 2 3 4) e B=(5 6 7 8 9)

Arquivo 12-lambda.rtk e 13-lambda.rtk

13. Execute os programas e indique o que faz cada um deles

12-lambda.rtk

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
;;
;;  Expressoes LAMBDA
;; ------------------------------------------------
;; Expressoes LAMBDA sao utilizados para criar novos procedimentos
;;    (lambda (var ....) expr1  expr2 ....)

;;------------ funcao Duas-vezes -------------

(define duasvezes
  (lambda (f x)   ;; <-------  2 parametros  f x
    (f x x)       ;; <-------  definicao da funcao
  )
 )
```

```
3+3 = 6
```

12-lambda-Romulo.rkt - DrRacket

```
;;------------ funcao com5 --------------

(define com5
  (lambda (operador x)
    (operador x 5)
  )
 )


;;------------ Polinomio P(x) = X^2 + 3X - 7 --------------

(define polinomio
  (lambda (x)                    ;<----- um parametro x
    (- (+ (* x x) (* 3 x)) 7)
  )
 )

;;-------- executando ... ---------------------------
;;---------------------------------------------------

(display "3+3 = ")
(duasvezes + 3)

(display "3*3 = ")
(duasvezes * 3)

(display "3-3 = ")
(duasvezes - 3)

(newline)
```

```
3+3 = 6
```

12-lambda-Romulo.rkt - DrRacket*

```
(display "3+3 = ")
(duasvezes + 3)

(display "3*3 = ")
(duasvezes * 3)

(display "3-3 = ")
(duasvezes - 3)

(newline)
(display "8+5 = ")
(com5 + 8)

(display "8*5 = ")
(com5 * 8)

(display "8-5 = ")
(com5 - 8)
(newline)

(display "P(x) = X^2 + 3X - 7   entao P(5) = ")
(polinomio 5)

(display "P(x) = X^2 + 3X - 7   entao P(0) = ")
(polinomio 0)

(display "P(x) = X^2 + 3X - 7   entao P(2) = ")
(polinomio 2)
```

```
3+3 = 6
```

```
(display "3+3 = ")
(duasvezes + 3)

(display "3*3 = ")
(duasvezes * 3)

(display "3-3 = ")
(duasvezes - 3)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes
3+3 = 6
3*3 = 9
3-3 = 0

8+5 = 13
8*5 = 40
8-5 = 3

P(x) = X^2 + 3X - 7   entao P(5) = 33
P(x) = X^2 + 3X - 7   entao P(0) = -7
P(x) = X^2 + 3X - 7   entao P(2) = 3
>
```

O programa acima realiza calculos de operações matemática, utilizando expressões Lambda, usada para criar novos procedimentos, assim realizando soma, subtração, multiplicação e até equações de segundo grau.

13-lambda.rtk

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
;;
;;  Expressoes LAMBDA e Recursão
;; ----------------------------------------------------
;; Expressoes LAMBDA sao utilizados para criar novos procedimentos
;;    (lambda (var ....) expr1  expr2 ....)


;; ------------ tamanho de uma lista ---------------
(define comprimento
  (lambda (lista)
    (if (null? lista)
        0
        (+ (comprimento (cdr lista)) 1)
        )
    )
  )
;;-------------- remove um elemento de uma lista -----
(define remove
  (lambda (x lista)
    (cond
```

```
Welcome to DrRacket, version 8.6 [cs].
```

```
    (cond
      ((null? lista) '())
      ((eqv? (car lista) x) (remove x (cdr lista)))
      (else (cons (car lista) (remove x (cdr lista))))
     )
    )
  )


;;-------------------------------------------------
(define inverso
  (lambda (n)
    (if (and (number? n) (not (= n 0)))
        (/ 1 n)
        "Nao é um numero"
     )
    )
  )



;; -------  Executando .....
;;-------------------------------------
;;
(define a (list 1 2 3 4 5 6 7))
(newline)
(display "lista: ")  a

 (display "tamanho da lista: ")
 (comprimento a)

 (display "Removendo um elemento (o 5): ")
 (remove 5 a)

 (display "Inverso de 12 = ")
 (inverso 12)
```

```
Welcome to DrRacket, version 8.6 [cs].
```

```
    (cond
      ((null? lista) '())
      ((eqv? (car lista) x) (remove x (cdr lista)))
      (else (cons (car lista) (remove x (cdr lista))))
     )
    )
  )


;;-------------------------------------------------
(define inverso
  (lambda (n)
    (if (and (number? n) (not (= n 0)))
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

lista: '(1 2 3 4 5 6 7)
tamanho da lista: 7
Removendo um elemento (o 5): '(1 2 3 4 6 7)
Inverso de 12 = 1/12
>
```

O programa acima utiliza funções para manipular listas e descobrir o comprimento, remover e inverter certos elementos definidos.

13.1. Crie um procedimento para realizar o cálculo de uma prestação em atraso, utilizando a fórmula Prest = valor + (valor*(taxa/100)*tempo). Dar exemplos.

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)
(define Prestacao 0)
(define tempo 12)
(define taxa 3)
(define valor 1000)
(set! Prestacao (+ valor (* valor (*(/ taxa 100) tempo))))
(display "Valor inicial: ")valor
(display "Meses de atraso: ")tempo
(display "Taxa(%): ")taxa
(display "Valor que deverá ser pago: ")Prestacao
```

```
Welcome to DrRacket. version 8.6 [cs].
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

Valor inicial: 3100
Meses de atraso: 2
Taxa(%): 3
Valor que deverá ser pago: 3286
>
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

Valor inicial: 1000
Meses de atraso: 12
Taxa(%): 3
Valor que deverá ser pago: 1360
>
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

Valor inicial: 50000
Meses de atraso: 1
Taxa(%): 3
Valor que deverá ser pago: 51500
>
```

13.2. O que faz o seguinte procedimento **abcd**:

```
(define abcd
(lambda (n)
(let f ((i 2))
(cond
((>= i n) '())
((integer? (/ n i))
(cons i (f (+ i 1))))
(else (f (+ i 1)))))))
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)
(define abcd
  (lambda (n)
    (let f ((i 2))
      (cond
        ((>= i n) '())
        ((integer? (/ n i))
         (cons i (f (+ i 1))))
        (else (f (+ i 1)))))))
(abcd 500)
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(newline)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

'(2 4 5 10 20 25 50 100 125 250)
>
```

O programa acima mostra a passagem de parametros de uma divisão por 2, sendo o valor passado por parâmetro maior ou igual a 2, além de mostrar na tela o dividendo da operação realizada

Arquivo 14-operad-logicos.rtk Operadores lógicos

14. Executar o programa e indicar o seu conteúdo

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
;;

(display "Operadores lógicos...   ")
(newline)

(let ((x 3))
  (and (> x 2) (< x 4)))

(let ((y 5))
  (and (> y 2) (< y 4)))
```

```
(newline)
(display "Tabela de valores AND")
(newline)
(display " T and T = ")  (and #t #t)
(newline)
(display " T and F = ")  (and #t #f)
(newline)
(display " F and F = ")  (and #f #f)

;;------------------
(newline)
(display "Tabela de valores OR")
(newline)
(display " T or T = ")   (or #t #t)
(newline)
(display " T or F = ")   (or #t #f)
(newline)
(display " F or F = ")   (or #f #f)
;;-----------
(newline)
(let ((x 3))
  (or (> x 2) (< x 8)))

(let ((y 10))
  (or (> y 20) (< y 7)))
;
(newline)
(display " not(5 > 3) = ")  (not (> 5 3))
(display " not(10 > 25) = ") (not (> 10 25))
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes
Operadores lógicos...
#t
#f

Tabela de valores AND
 T and T = #t

 T and F = #f

 F and F = #f

Tabela de valores OR
 T or T = #t

 T or F = #t

 F or F = #f

#t
#f

 not(5 > 3) = #f
 not(10 > 25) = #t
>
```

O programa acima mostra comparações entre operadores lógicos AND, que retorna um BOOL True, caso as duas condições sejam verdadeiras e OR, retornando verdade se uma ou mais expressões forem verdadeiras.

14.1. Escreva e teste pelo menos cinco operações lógicas

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
(let ((a 15))
  (and (< a 10) (< a 20)))

(let ((b 15))
  (or (< b 10) (< b 20)))

(let ((c 5))
  (and (< c 10) (>= c 2)))

(let ((d 2))
  (or (>= d 2) (> d 5)))

(let ((e 1))
  (and (<= e 2) (> e 0)))
```

```
  Aluno:  Rômulo Souza Fernandes
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes
#f
#t
#t
#t
#t
>
```

Arquivo 15-predicados.rtk Predicados

15. Executar o programa e indicar o seu conteúdo

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
;; Predicados
;; ------------------------------------------------
(display "Operador IGUAL ")
(newline)
(display "a = 3? ") (eq? 'a 3)
(display "4 = 4? ") (eq? 4 4)
(display "car = car? ") (eq? car car)
(display "car  = cdr? ")(eq? car cdr)
(display "Falso = Falso? ")(eq? #f #f)
;;

(newline)
(display "Operador EQUIVALENTE ")
(newline)
(display "9/2 é EQUIVALENTE a 7/2? ")(eqv? 9/2 7/2)
(display "12/5 é EQUIVALENTE a 24/10? ")(eqv? 12/5 24/10)

;;
;;
(let ((x  "Tudo bem?")) (eqv? x x))
;;
(boolean? #t)
```

```
Par (cons 6 9) ?:  #t
```

```
(boolean? #f)
(display "7 é BOOLEANO? ")(boolean? 7)
;;
;; Lista vazia
(display "( ) é uma lista NULA? ")(null?'())
(display "(a) é uma lista NULA? ")(null? '(a))
(display "(list 1 3 5 7) é uma lista NULA? ")(null? (list 1 3 5 7))
(null? 6)
;;
(display "Pares?:  ") (newline)
(display "Par (3.4) ?:  ")(pair? '(3 . 4))
(display "Par 5?:  ")(pair? 5)
(display "Par (cons 6 9) ?:  ") (pair? (cons 6 9))
;;
(display "Numero 6.78 ?:  ")(number? 6.78)
(display "Numero 2/7 ?:  ")(number? 2)
;
(newline)
(display "É un número INTEIRO 3 ?:  ")(integer? 3)
(display "É un número INTEIRO 3.0 ?:  ")(integer? 3.0)
(display "É un número INTEIRO 3/7 ?:  ")(integer? 3/7)
;;

(newline)
(display "É un número REAL 2 ?:  ")(real? 2)
(display "É un número REAL 3/5 ?:  ")(real? 3/5)
(display "É un número REAL 4+5i ?:  ")(real? 4+5i)

;;
(newline)
(display "É un número COMPLEXO 5+3i ?:  ")(complex? 5+3i)
;;

(newline)
(display "É un STRING 'Oi, UENF ...' ?:  ")(string? "Oi, UENF ...")
(display "É un STRING 24 ?:  ")(string? 24)
```

```
Par (cons 6 9) ?:  #t
```

```
38 | (boolean? #f)

Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes
Operador IGUAL
a = 3? #f
4 = 4? #t
car = car? #t
car  = cdr? #f
Falso = Falso? #t

Operador EQUIVALENTE
9/2 é EQUIVALENTE a 7/2? #f
12/5 é EQUIVALENTE a 24/10? #t
#t
#t
#t
7 é BOOLEANO? #f
( ) é uma lista NULA? #t
(a) é uma lista NULA? #f
(list 1 3 5 7) é uma lista NULA? #f
#f
Pares?:
Par (3.4) ?:  #t
Par 5?:  #f
Par (cons 6 9) ?:  #t
Numero 6.78 ?:  #t
Numero 2/7 ?:  #t

É un número INTEIRO 3 ?:  #t
É un número INTEIRO 3.0 ?:  #t
É un número INTEIRO 3/7 ?:  #f

É un número REAL 2 ?:  #t
É un número REAL 3/5 ?:  #t
É un número REAL 4+5i ?:  #f

É un número COMPLEXO 5+3i ?:  #t

É un STRING 'Oi, UENF ...' ?:  #t
É un STRING 24 ?:  #f
>
```

O programa acima utiliza predicados que retornam BOOL depois de serem usados para a comparação necessária, é possível unir o operador com as funções mostradas nos programas anteriores, exemplo, comparação de string e verificação do tipo primitivo de uma determinada variável.

15.1. Testar os predicados: (char? ´m) , (char? 14), (char? #\b) , (char? #\m)

```
1  ;; Introdução à Linguagem Scheme-Racket
2  ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
3  ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4  ;; Setembro - 2022
5  ;; Aluno: Rômulo Souza Fernandes
6  ;;
7  #lang racket      ;; define a linguagem default
8  ; ----------------------------------------------
9  (display "  UENF-CCT-LCMAT-CC, 2022")
10 (newline)
11 (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display "  Aluno:  Rômulo Souza Fernandes ")
14 (newline)
15 (display "É um Char m? ") (char? 'm)
16 (display "É um Char 14? ") (char? 14)
17 (display "É um Char #\\b? ") (char? #\b)
18 (display "É um Char #\\m ? ") (char? #\m)

Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes
É um Char m? #f
É um Char 14? #f
É um Char #\b? #t
É um Char #\m ? #t
>
```

Arquivo 16-mapeamentos.rtk Mapeamentos

15.2. Executar o programa e indicar o que ele faz

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
;;
;; Mapeamentos
; ----------------------------------------------

(define quadrado
  (lambda ( x ) (* x x)
  )
)
;;;;
(define dobro
  (lambda (y) (* 2 y)
  )
)
;;;;
(define proximo
  (lambda ( x ) (+ x 1)
```

Welcome to DrRacket, version 8.6 [cs].

```
)
;;;;
(define proximo
  (lambda ( x ) (+ x 1)
  )
)

;;
;;

(define Lista (list  1 4 9 16 25))
(display "Lista:  ") Lista
(display "Raiz Lista:  ") (map sqrt Lista)
;;
;;
(newline)
(define dados '(2 5 6 23))
(newline)
(display "Dados:  ") dados
(newline)
;;mapeamentos
(display "Mapeamentos: Dados-Proximo  ") (newline)
(map proximo dados)

(display "Mapeamentos: Dados-Quadrado  ") (newline)
(map quadrado dados)

(display "Mapeamentos: Dados-Dobro  ") (newline)
(map dobro dados)
```

Welcome to DrRacket, version 8.6 [cs].

```
29  )
30  ;;;;
31  (define proximo
32    (lambda ( x ) (+ x 1)
33    )
34  )
35
36  ;;
37  ;;
38
39  (define Lista (list  1 4 9 16 25))
40  (display "Lista:  ") Lista
41  (display "Raiz Lista:  ") (map sqrt Lista)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes
Lista:  '(1 4 9 16 25)
Raiz Lista:  '(1 2 3 4 5)


Dados:  '(2 5 6 23)

Mapeamentos: Dados-Proximo
'(3 6 7 24)
Mapeamentos: Dados-Quadrado
'(4 25 36 529)
Mapeamentos: Dados-Dobro
'(4 10 12 46)
>
```

O programa acima demonstra a função map, que possui a função de percorrer uma lista, com base em uma função já definida.

15.3. Construir um NOVO programa que faça o seguinte mapeamento

$$x \longrightarrow x^2 + 3x - 9$$

Arquivo 17-raizes-poly.rtk Aplicações: Raízes de polinômios

16. Executar o programa e explicar o conteúdo e os resultados

```
1   ;; Introdução à Linguagem Scheme-Racket
2   ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
3   ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4   ;; Setembro - 2022
5   ;; Aluno: Rômulo Souza Fernandes
6   ;;
7     ;; define a linguagem default ===> Habilite Advanced Student
8   ; ------------------------------------------------
9   (newline)
10  (display "  UENF-CCT-LCMAT-CC, 2022")
11  (newline)
12  (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
13  (newline)
14  (display "  Aluno:  Rômulo Souza Fernandes ")
15  (newline)
16  ;;;
17  ;;
18  ;;  Aplicações:  raizes do polinomio Ax^2 + Bx + C = 0
19  ;; ------------------------------------------------
20
21  (define (poly2grau a b c)
22    (cond
23      [(= a 0) "degenerada" ]
24      [(< (* b b)
25          (* 4 a c))
26       "Nenhuma Ou Complexa" ]
27      [(= (* b b)
28          (* 4 a c))
29       (/ (- b) (* 2 a))]
30      [(> (* b b)
31          (* 4 a c))
32       (list (/ (+ (- b)
33                   (sqrt (- (* b b) (* 4 a c))))
34                (* 2 a))
35             (/ (- (- b)
36                   (sqrt (- (* b b) (* 4 a c))))
```

```
39 ;;-------------------------------------
40 (newline)
41 ;; EXEMPLOS
42 (display "X^2 + 2X + 1 = 0 , Raizes = ")
43 (poly2grau 1 2 1)
44 (display "deberia ser -1")
45 (newline)
46
47 (display "3X^2 + 4X + 1 = 0 , Raizes = ")
48 (poly2grau 3 4 1)
49 (display "deberia ser -1/3 -1")
50 (newline)
51
52 (newline)
53 (display "2X^2 + 4X + 3 = 0 , Raizes = ")
54 (poly2grau 2 4 3)
55 (display "deberia ser Nenhuma")
56 (newline)
57
58 (newline)
59 (display "X^2 -1 = 0 , Raizes = ")
60 (poly2grau 1 0 -1)
61 (display "deberia ser 1 e -1")
62 (newline)
63
64 (newline)
65 (display "2X^2 +4X +2 = 0 , Raizes = ")
66 (poly2grau 2 4 2)
67 (display "deberia ser -1")
68 (newline)
69
70 (newline)
71 (display "X + 1 = 0 , Raizes = ")
72 (poly2grau 0 1 1)
73 (display "deberia ser: Degenerada")
74 (newline)
```

```
39 ;;-------------------------------------
40 (newline)
41 ;; EXEMPLOS
42 (display "X^2 + 2X + 1 = 0 , Raizes = ")
43 (poly2grau 1 2 1)
44 (display "deberia ser -1")
45 (newline)
46
47 (display "3X^2 + 4X + 1 = 0 , Raizes = ")
48 (poly2grau 3 4 1)
49 (display "deberia ser -1/3 -1")
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student [custom]; memory limit: 128 MB.
Teachpack: gui.rkt.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

X^2 + 2X + 1 = 0 , Raizes = -1
deberia ser -1
3X^2 + 4X + 1 = 0 , Raizes = (list -0.3̄ -1)
deberia ser -1/3 -1

2X^2 + 4X + 3 = 0 , Raizes = "Nenhuma Ou Complexa"
deberia ser Nenhuma

X^2 -1 = 0 , Raizes = (list 1 -1)
deberia ser 1 e -1

2X^2 +4X +2 = 0 , Raizes = -1
deberia ser -1

X + 1 = 0 , Raizes = "degenerada"
deberia ser: Degenerada
>
```

O programa acima calcula as raízes de polinômios, se é complexa ou degenerada, além de informar qual deveria ser a resposta correta, assim garantindo que tudo está funcionando corretamente.

16.1. Fazer testes para outros cincos polinômios de segundo grau

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera        (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
  ;; define a linguagem default ===> Habilite Advanced Student
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;;
;;
;;  Aplicações:  raizes do polinomio Ax^2 + Bx + C = 0
;; ----------------------------------------------

(define (poly2grau a b c)
  (cond
    [(= a 0) "degenerada" ]
    [(< (* b b)
        (* 4 a c))
     "Nenhuma Ou Complexa" ]
    [(= (* b b)
        (* 4 a c))
     (/ (- b) (* 2 a))]
    [(> (* b b)
        (* 4 a c))
     (list (/ (+ (- b)
                 (sqrt (- (* b b) (* 4 a c))))
              (* 2 a))
           (/ (- (- b)
```

Welcome to DrRacket, version 8.6 [cs].

```
                 (sqrt (- (* b b) (* 4 a c))))
              (* 2 a)))]))

;;-----------------------------------
(newline)
;; EXEMPLOS
(display "X^2 - 4X + 5 = 0 , Raizes = ")
(poly2grau 1 -4 5)
(display "Deveria ser Nenhuma ou Complexa")
(newline)

(newline)
(display "4X^2 - 4X + 1 = 0 , Raizes = ")
(poly2grau 4 -4 1)
(display "Deveria ser 1/2")
(newline)

(newline)
(display "X^2 - 5X + 6 = 0 , Raizes = ")
(poly2grau 1 -5 6)
(display "Deveria ser 3 e 2")
(newline)

(newline)
(display "5x^2 - 45x = 0 , Raizes = ")
(poly2grau 5 -45 0)
(display "Deveria ser 9 e 0")
(newline)

(newline)
(display "x^2 , Raizes = ")
(poly2grau 1 0 0)
(display "Deveria ser 0")
(newline)
```

Welcome to DrRacket, version 8.6 [cs].

```
                    (sqrt (- (* b b) (* 4 a c))))
                (* 2 a)))]))

;;-------------------------------------
(newline)
;; EXEMPLOS
(display "X^2 - 4X + 5 = 0 , Raizes = ")
(poly2grau 1 -4 5)
(display "Deveria ser Nenhuma ou Complexa")
(newline)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

X^2 - 4X + 5 = 0 , Raizes = "Nenhuma Ou Complexa"
Deveria ser Nenhuma ou Complexa

4X^2 - 4X + 1 = 0 , Raizes = 0.5
Deveria ser 1/2

X^2 - 5X + 6 = 0 , Raizes = (list 3 2)
Deveria ser 3 e 2

5x^2 - 45x = 0 , Raizes = (list 9 0)
Deveria ser 9 e 0

x^2 , Raizes = 0
Deveria ser 0
>
```

Arquivo 20-estruturas.rtk Aplicações: Estruturas de dados

17. Executar o programa e explicar o conteúdo e os resultados

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
  ;; define a linguagem default ===> Habilite Advanced Student
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;;
; Ir no Menu:
;   Language -> Choose Language -> How to Design Programs -> Advanced Student
;-----------------------------------------------------------------------------

;;
(define-struct entrada (nome cep fone))

(newline)
(make-entrada 'PedroSilva '05890-001 '606-7771)
(newline)
(display "Nome = " )
(entrada-nome (make-entrada 'PedroSilva '05890-001 '606-7771))
(display "CEP = " )
(entrada-cep (make-entrada 'PedroSilva '05890-001 '606-7771))
;;
;;
(define abc (make-entrada 'PedroSilva '05890-001 '606-7771))
(display "Fone = " )
(entrada-fone abc)
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
  ;; define a linguagem default ===> Habilite Advanced Student
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

(make-entrada 'PedroSilva '05890-001 '606-7771)

Nome = 'PedroSilva
CEP = '05890-001
Fone = '606-7771
>
```

O programa acima utiliza a estrutura struct para armazenar dados de um usuário, como nome, cep e telefone. Ocorre a entrada de dados e por fim esses dados são printados na tela.

17.1. Em um novo programa defina pelo menos outras TRÊS estruturas diferentes

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
  ;; define a linguagem default ===> Habilite Advanced Student
; ----------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;;
; Ir no Menu:
;   Language -> Choose Language -> How to Design Programs -> Advanced Student
;-----------------------------------------------------------------------------

;;
(define-struct dados (nomeCompleto rg cpf))

(newline)
(make-dados 'AlexSeverinoPereira '1234567 '12345671234)
(newline)

(display "Nome Completo = " )
(dados-nomeCompleto (make-dados 'AlexSeverinoPereira '1234567 '12345671234))

(display "RG = " )
(dados-rg (make-dados 'AlexSeverinoPereira '1234567 '12345671234))
```

```
Welcome to DrRacket, version 8.6 [cs].
```

```
(display "CPF = " )
(dados-cpf (make-dados 'AlexSeverinoPereira '1234567 '12345671234))

;;

(define-struct notas (p1 p2 p3))

(newline)
(make-notas '5 '9 '6)
(newline)

(display "Nota da p1 = " )
(notas-p1 (make-notas '5 '9 '6))

(display "Nota da p2 = " )
(notas-p2 (make-notas '5 '9 '6))

(display "Nota da p3 = " )
(notas-p3 (make-notas '5 '9 '6))

;;

(define-struct aluno (nome matricula media))

(newline)
(make-aluno 'JorgeBaroneLopes '0011223344 '8)
(newline)

(display "Nome do aluno = " )
(aluno-nome (make-aluno 'JorgeBaroneLopes '0011223344 '8))

(display "Matricula do aluno= " )
(aluno-matricula (make-aluno 'JorgeBaroneLopes '0011223344 '8))

(display "Média do aluno = " )
(aluno-media (make-aluno 'JorgeBaroneLopes '0011223344 '8))
```

```
Welcome to DrRacket, version 8.6 [cs].
```

```
(dados-cpf (make-dados 'AlexSeverinoPereira '1234567 '12345671234))

;;

(define-struct notas (p1 p2 p3))

(newline)
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes

(make-dados 'AlexSeverinoPereira 1234567 12345671234)

Nome Completo = 'AlexSeverinoPereira
RG = 1234567
CPF = 12345671234

(make-notas 5 9 6)

Nota da p1 = 5
Nota da p2 = 9
Nota da p3 = 6

(make-aluno 'JorgeBaroneLopes 11223344 8)

Nome do aluno = 'JorgeBaroneLopes
Matricula do aluno= 11223344
Média do aluno = 8
>
```

Arquivo 30-entrada.rtk

18. Executar o programa e explicar o conteúdo e os resultados

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
(newline)
(newline)

(display "Escreva tres notas (numeros menores que 10, separados por espaços a  b  c) : ")
(newline)
(define a (read))
(define b (read))
(define c (read))


(define media
  (lambda (x y z)
  (/ (+ (+ x y) z) 3)
    ))

(display "A media = ")
(media a b c)

(if (> (media a b c) 5)
    (display "Aprovado")
    (display "Reprovado"))
;;----------------------------------------
;; Defina aqui una media com notas com pesos diferentes
```

```
Welcome to DrRacket, version 8.6 [cs].
```

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
(newline)
(newline)

(display "Escreva tres notas (numeros menores que 10, separados por espaços a  b  c) : ")
(newline)
(define a (read))
(define b (read))
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno:  Rômulo Souza Fernandes


Escreva tres notas (numeros menores que 10, separados por espaços a  b  c) :
```

O programa acima começa a utilizar dados informados pelo usuário, utilizando o (read), após isso ele calcula a média e informa a o usuário está aprovado ou reprovado, utilizando if.

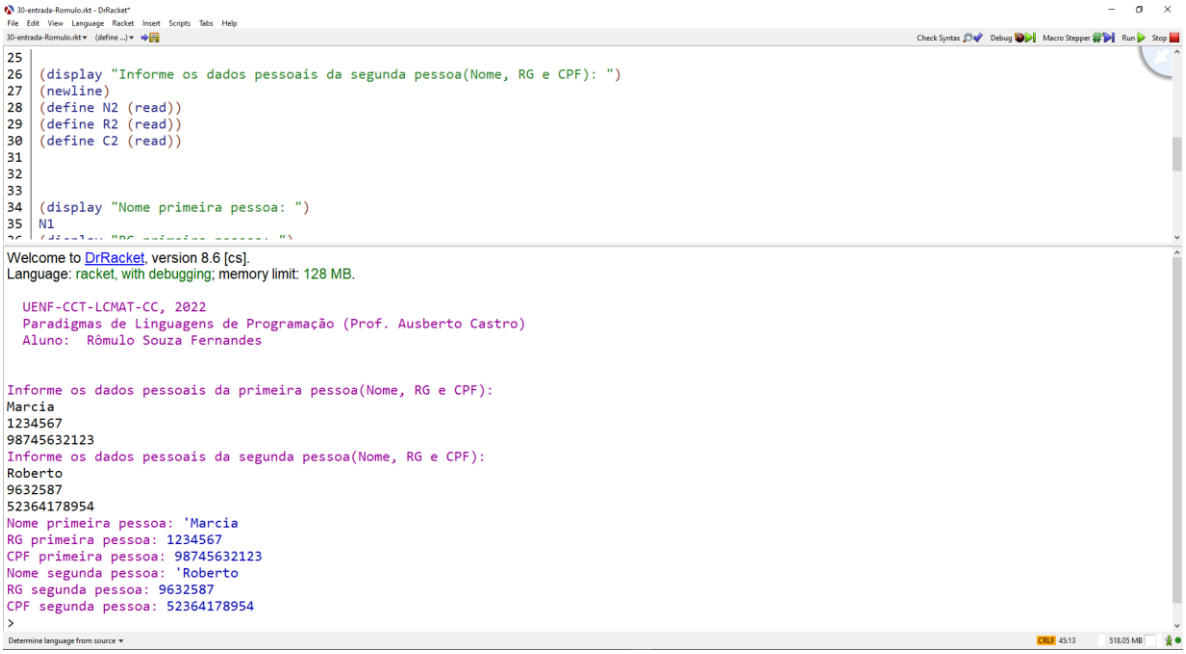
18.1. Criar um programa NOVO que faça a leitura de dados pessoais de duas pessoas (utilize entrada de dados)

```
;; Introdução à Linguagem Scheme-Racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; Setembro - 2022
;; Aluno: Rômulo Souza Fernandes
;;
#lang racket      ;; define a linguagem default
; ------------------------------------------------
(newline)
(display "  UENF-CCT-LCMAT-CC, 2022")
(newline)
(display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
(newline)
(newline)

(display "Informe os dados pessoais da primeira pessoa(Nome, RG e CPF): ")
(newline)
(define N1 (read))
(define R1 (read))
(define C1 (read))

(display "Informe os dados pessoais da segunda pessoa(Nome, RG e CPF): ")
(newline)
(define N2 (read))
(define R2 (read))
(define C2 (read))

(display "Nome primeira pessoa: ")
N1
```

Welcome to DrRacket, version 8.6 [cs].

```
(newline)
(display "  Aluno:  Rômulo Souza Fernandes ")
(newline)
;;
(newline)
(newline)

(display "Informe os dados pessoais da primeira pessoa(Nome, RG e CPF): ")
(newline)
(define N1 (read))
(define R1 (read))
(define C1 (read))

(display "Informe os dados pessoais da segunda pessoa(Nome, RG e CPF): ")
(newline)
(define N2 (read))
(define R2 (read))
(define C2 (read))

(display "Nome primeira pessoa: ")
N1
(display "RG primeira pessoa: ")
R1
(display "CPF primeira pessoa: ")
C1

(display "Nome segunda pessoa: ")
N2
(display "RG segunda pessoa: ")
R2
(display "CPF segunda pessoa: ")
C2
```

Welcome to DrRacket, version 8.6 [cs].



Parte 2:

Resolver a lista de Exercícios no final da Notas de Aula (Slide 50)