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

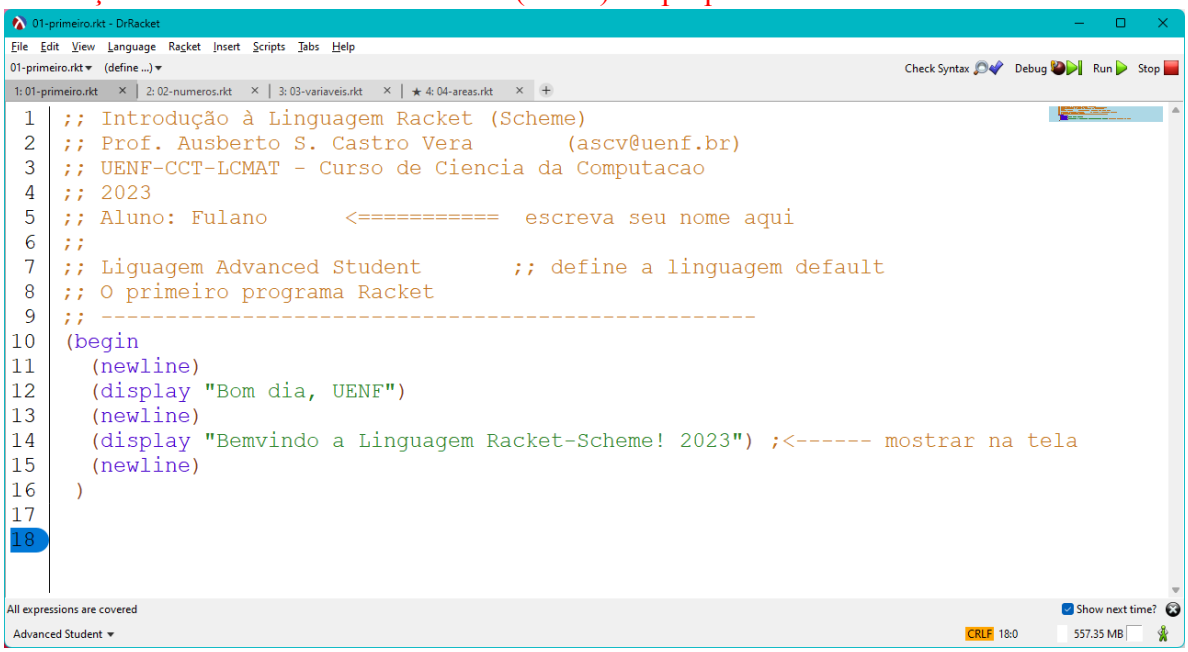
Data: 10 de abril de 2023

Prática – Racket

Nome Completo: Gabriel Costa Fassarella
Data: 26 de abril de 2023 Total Exercícios Resolvidos: .....

Arquivo 01-primeiro.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 7 linhas (10-16). Explique cada uma delas?



Codigo fonte

Interações (shell)

Explicar aqui: A linha 10 apresenta o comando "(begin" que indica o início do código, as linhas 11, 13 e 15 apresentam o comando "(newline)", que indica uma quebra de linhas, e as linhas 12 e 14 apresentam o comando "(display)" seguido de uma string no interior dos parênteses, indicando que essa string será mostrada na tela.

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

```
1  ;; Introdução à Linguagem Racket (Scheme)
2  ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
3  ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4  ;; 2023
5  ;; Aluno: Gabriel Costa Fassarella      <===========  escreva seu nome aqui
6  ;;
7  ;; Liguagem Advanced Student       ;; define a linguagem default
8  ;; O primeiro programa Racket
9  ;; ------------------------------------------------
10 (begin
11   (newline)
12   (display "Bom dia, UENF 30 anos")
13   (newline)
14   (display "Bemvindo a Linguagem Racket-Scheme! 2023") ;<------ mostrar na tela
15   (newline)
16   (display "Pratica 01 - Linguagem Racket")
17   (newline)
18   (display "Gabriel Costa Fassarella")
19   (newline)
20   (display "11/04/2023")
21   (newline)
22  )
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.

Bom dia, UENF 30 anos
Bemvindo a Linguagem Racket-Scheme! 2023
Pratica 01 - Linguagem Racket
Gabriel Costa Fassarella
11/04/2023
```

Arquivo 02-numeros.rtk Números e Aritmética

3. Execute o programa e mostre os resultados

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
    UENF-CCT-LCMAT-CC, 2023
    Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
    Aluno:  Gabriel Costa Fassarella
Soma 11 + 35 = 46
Produto 23*14 = 322
Mistura 7 + (3*9) = 34
Combinando (2 + (3*4))/2 - 4 = 3
Raiz quadrada de 4= 2
Raiz quadrada de 7= #i2.6457513110645907
Complexos - raiz quadrada de -1= 0+1i

Valor de Pi+7 #i10.141592653589793
Seno 60 graus: #i0.8660254037844386
Coseno 60 graus: #i0.5000000000000001
Coseno 45 graus: #i0.7071067811865476
Logaritmo Natural de 17: #i2.833213344056216
exponente 8^3 = 512
exponente 4^(1/2) = 2
Maximo de  1 32 40 27 3 = 40
minimo de  11 3 41 25 3 = 3
valor absoluto de 11 = 11
valor absoluto de -7 = 7

"quotes obriga as listas serem tratadas como DADOS"
(list 5 1 9)
(list '/ 4 (list '* 3 7))
(list 1 'a 3 'b)
(list (list 'a 'b) (list 3 5))
(list '+ 2 (list '* 5 7))
>
```

4. Escreva programas Racket para as seguintes expressões:
   4.1. P = $(6 - (8^2 - 4^3) / 6) - (7 + (2 - (3^4 - 5)$

```
;; Introdução à Linguagem Racket (Scheme)
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2023
;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui
;;
;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;

; -----------------------------------------------
(display "Resultado da expressão")
(- (- 6 (/ (- (* 8 8) (* 4 (* 4 4))) 6)) (+ 7 (- 2 (- (* 3 3 (* 3 3)) 5))))
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
Resultado da expressão73
>
```

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

$$\frac{\sqrt{5^2 + 6 + sin(12 - 8) + cos(20 + 25)}}{(5 - 3) * (4 + 8)^2}$$

```
;; Introdução à Linguagem Racket (Scheme)
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2023
;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui
;;
;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;

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

(display "Resultado da expressão: ")
(/ ( sqrt ( + ( + ( * 5 5) 6 ) ( + ( sin ( - 12 8 ))) ( cos ( + 20 25)) ) ) ( * ( - 5 3 ) ( * ( + 4 8) ( + 4 8) )
```

```
Benvindo a DrRacket, versão 7.2 [3m].
Linguagem: Advanced Student; memory limit: 128 MB.
Resultado da expressão: #i0.01926020086234269
>
```

Arquivo 03-variaveis.rtk

5. Execute o programa e indique o valor das variáveis m, z, k
   5.1. Explicar o significado de cada uma das 3 expressões de iteração `let`
   A expressão (let ((x 24)) altera o valor de x para 24 temporariamente, somando assim o valor de x com 6. O mesmo ocorre na segunda expressão, alterando o valor de a e b. Já a terceira ele muda o valor de op1 e op2 para uma operação, e o de x para um valor.
   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
;; 2023
;; Aluno: Gabriel Costa Fassarella      <===========  escreva seu nome aqui

;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
;;
; -----------------------------------------------

(display "14 + 46 = ")
(let ((x 14) (y 46) (op +)) ;; Declarando x = 14, y = 46 e op = +
  (op x y)) ;; realizando a operação + 14 46 <=> 14 + 46 = 60
(newline)

(display "(37 + 93) - 17 = ")
(let ((x 37) (y 93) (z 17) (op1 +) (op2 -))
    (op2 (op1 x y) z))
(newline)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
14 + 46 = 60

(37 + 93) - 17 = 113
```

Arquivo 04-areas.rtk

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

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 Racket (Scheme)
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2023
;; Aluno: Gabriel Costa Fassarella      <===========  escreva seu nome aqui
;;
;; Liguagem Advanced Student       ;; define a linguagem default
;; O primeiro programa Racket
;; -----------------------------------------------------

(display "Area do quadrado: ")
(let ((l 5))
    (* l l))
(newline)

(display "Area de um trapezio: ")
(let ((b 5) (B 10) (h 4))
    (/ (* (+ b B) h) 2))
(newline)

(display "Area de um poligono: ")
(let ((ql 4) (l 5))
  (/ (* (* ql l) (/ l 2)) 2))
(newline)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
Area do quadrado: 25

Area de um trapezio: 30

Area de um poligono: 25
>
```

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

```
1  ;; Introdução à Linguagem Racket (Scheme)
2  ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
3  ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4  ;; 2023
5  ;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui
6  ;;
7  ;; Liguagem Advanced Student       ;; define a linguagem default
8  ;; O primeiro programa Racket
9  ;; ------------------------------------------------
10
11 (define Pi 3.141593)
12
13 (display "Volume: ")
14 (let ((r 10) (a 30))
15     (* Pi (* (* r r) a)))
16 (newline)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
Volume: 9424.779
```

Arquivo 05-funcoes.rtk

7. Execute o programa e explique o que faz

   O programa em questão apresenta inúmeras funções sendo declaradas utilizando o comando (define). Um exemplo é a primeira função declarada, a "ADICIONA", que é apresentada por meio do (define), em seguida o nome da função, seguido dos parâmetros e por fim o corpo com o procedimento. No caso da função "quadrado", ela é definida por meio do (define quadrado), apresentando logo em seguida o seu parâmetro, nesse caso o "a", e logo após um (display) mostrando um texto seguido do cálculo desejado, nesse caso o quadrado de a, indicado por (* a a). Essas funções são chamadas a partir da linha 89, onde é chamada pelo seu nome e pelos parâmetros necessários dentro de parênteses.

   7.1. Escreva um NOVO programa Racket onde é definido duas funções (procedimentos lambda)

```
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  ;; 2023
5  ;; Aluno: Gabriel Costa Fassarella     <===========  escreva seu nome aqui
6
7  ;;;;;;;;;;;;;;;   Escolha a linguagem R5RS
8  ; -----------------------------------------------
9
10 (define raiz
11   (lambda (x)
12     (begin
13       (display "Raiz de ")
14       (display x)
15       (display " = ")
16       (display (sqrt x))
17       (newline))))
18
19 (define seno
20   (lambda (x)
21     (begin
22       (display "Seno(") (display x) (display ") = ")
23       (sin x)
24       )
25     )
26   )
27
28 (raiz 25)
29 (seno 90)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
Raiz de 25 = 5
Seno(90) = #i0.8939966636005579
```

Arquivo 06-condicionalIF.rtk

8. Executar e explicar o programa

   O código em questão apresenta algumas estruturas condicionais. Na linha 22, é iniciado um P com valor 528, que passa por um if que verifica se P é maior ou menor que 57, neste caso maior. Em seguida, é iniciada uma função chamada reply, que recebe uma string que verifica se ela começa com “oi”, se recomeçar retorna “tudo bem”, caso contrário retorna “não entendi”. Logo após, existe uma função chamada miniquadrado que calcula o quadrado de um número caso ele seja menor que um segundo número.

   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
;; 2023
;; Aluno:   Gabriel Costa Fassarella    <===========  escreva seu nome aqui

;;;;;;;;;;;;;;;;   Escolha a linguagem R5RS
;;
;; Ajuda:  http://docs.racket-lang.org/guide/syntax-overview.html#(part._.Conditionals_with_if__and__or__and_cond)
; -----------------------------------------------------

(define menab
  (lambda(a b)
    (begin
      (if (< a b)
          (display "a é menor que b")
          (display "b é menor que a")
          )
      )
    )
  )

(define menabc
  (lambda(a b c)
    (begin
      (if (< (+ a b) c)
          (display "a+b é menor que c")
          (display "a+b é maior que c")
      )
    )
  )
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
a é menor que b
a+b é maior que c
```

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
;; 2023
;; Aluno:   Gabriel Costa Fassarella    <===========  escreva seu nome aqui

;;;;;;;;;;;;;;;;   Escolha a linguagem R5RS
;;
;; Ajuda:  http://docs.racket-lang.org/guide/syntax-overview.html#(part._.Conditionals_with_if__and__or__and_cond)
; -----------------------------------------------------

(define media
  (lambda (p1 p2 p3)
    (begin
      (if (>= (/ (+ (+ p1 p2) p3) 3) 6)
          (display "Aprovado")
          (display "Reprovado")
          )
      )
  )
 )

(media 3 6 9)
(newline)
(media 6 2 4)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
Aprovado
Reprovado
>
```

Arquivo 07-formulas.rtk

9. Executar e explicar o programa

   O programa em questão apresenta inúmeras funções responsáveis por calcular fórmulas matemáticas, dentre elas a de Pitágoras, feita por meio de uma linha de código que indica uma soma e uma potência de 2 números. Existem também a fórmula da aproximação, que utiliza o método recursivo para tal, ou seja, ele apresenta um caso

base, que funcionará como caso de saída, e existe um caso geral, no qual ele chamará a função novamente, esses casos são verificados por meio de estruturas condicionais. O mesmo acontece com a fórmula de fatorial, que também utiliza um método recursivo para tal.

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

```
1  (define (fatorial n)
2    (cond
3      ((= n 0) 1)
4      ((< n 0) "Não existe fatorial de número negativo.")
5      (else (* n (fatorial (- n 1))))))
6
7  (fatorial 5)
8
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
120
>
```

Arquivo 08-condicional.rtk

10. Execute o programa e indique o que faz

O código verifica na função “taxa quantidade” 3 condições, se uma delas for verificada, o valor correspondente será retornado, caso nenhuma das condições sejam satisfeitas será retornado o valor 1.

10.1. Escreva um programa condicional com pelo menos 5 opções

```
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  ;; Abril 2023
5  ;; Aluno: Gabriel Costa Fassarella     <===========  escreva seu nome aqui
6  ;;
7  ;; #lang racket      ;; define a linguagem default
8  ; ------------------------------------------------
9
10 (define (verif-num valor)
11   (cond
12     ((< valor 0) "Valor é menor que zero")
13     ((= valor 0) "O numéro é igual a 0")
14     ((and (> valor 0) (< valor 50)) "O número é menor que 50")
15     ((and (>= valor 10) (< valor 50)) "Valor se encontra entre 10 e 50")
16     ((>= valor 50) "O número é maior ou igual a 50")
17     (else "Valor inválido")))
18
19 (verif-num 15)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
"Valor é menor que 50"
>
```

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

```
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  ;; Abril 2023
5  ;; Aluno: Gabriel Costa Fassarella     <===========  escreva seu nome aqui
6  ;;
7  ;;#lang racket      ;; define a linguagem default
8  ; ------------------------------------------------
9
10 (define (bhaskara1 a b c)
11   (begin
12     (display "x1 = ")
13     (let ((delta (-(expt b 2) (* 4 (* a c)))))
14         (if (< delta 0)
15             (error "delta menor que 0...")
16             (/ (- (- b) (sqrt delta)) (* 2 a)))
17         )
18   )
19 )
20
21 (define (bhaskara2 a b c)
22   (begin
23     (display "x2 = ")
24     (let ((delta (-(expt b 2) (* 4 (* a c)))))
25         (if (< delta 0)
26             (error "delta menor que 0...")
27             (/ (+ (- b) (sqrt delta)) (* 2 a)))
28         )
29   )
30 )
31
32 (bhaskara1 22 -55 10)
33 (newline)
34 (bhaskara2 22 -55 10)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
x1 = #i0.19740580209914443

x2 = #i2.3025941979008553
```

Arquivo 09-pares.rtk

11. Execute o programa

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

```
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  ;; Abril 2023
5  ;; Aluno: Gabriel Costa Fassarella      <===========  escreva seu nome aqui
6  ;;
7  #lang racket      ;; define a linguagem default
8  ; -----------------------------------------------
9
10 (define p1 (cons 15 105))
11 (define p2 (cons 'k 62.5))
12
13 (display "Primeiro elemento de p1 = ")
14 (display (car p1))
15 (newline)
16 (display "Segundo elemento de p1 = ")
17 (display (cdr p1))
18 (newline)
19
20 (display "Primeiro elemento de p2 = ")
21 (display (car p2))
22 (newline)
23 (display "Segundo elemento de p2 = ")
24 (display (cdr p2))
25 (newline)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
Primeiro elemento de p1 = 15
Segundo elemento de p1 = 105
Primeiro elemento de p2 = k
Segundo elemento de p2 = 62.5
>
```

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

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

    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

```
1  ;; Introdução à Linguagem Racket (Scheme)
2  ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
3  ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4  ;; 2023
5  ;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui
6  ;;
7  #lang racket      ;; define a linguagem default
8  ;; define a linguagem default: R5RS
9  ; ------------------------------------------------
10
11 (define lista1 (list 1 2 3 4 5 6))
12 (define lista2 (list 7 8))
13
14 (display "lista 1 = ")
15 (display lista1)
16 (newline)
17 (display "Primeiro item = ")
18 (display (car lista1))
19 (newline)
20 (display "Ultimo item = ")
21 (display (car(reverse lista1)))
22 (newline)
23 (display "Comprimento da Lista = ")
24 (display (length lista1))
25 (newline)
26 (display "lista 1 + lista 2 = ")
27 (display (append lista1 lista2))
28
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
lista 1 = (1 2 3 4 5 6)
Primeiro item = 1
Ultimo item = 6
Comprimento da Lista = 6
lista 1 + lista 2 = (1 2 3 4 5 6 7 8)
>
```

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

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.

```
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  ;; 2023
5  ;; Aluno: Gabriel Costa Fassarella      <===========  seu nome aqui e abaixo
6  ;;
7  #lang racket      ;; define a linguagem default
8  ; -----------------------------------------------
9
10 (define prestacao
11   (lambda (val tax t)
12     (display "Valor prestação = ")
13     (display (+ val (* (* val (/ tax 100)) t)))
14     (newline)
15     )
16   )
17
18 (prestacao 1000 10 5)
19 (prestacao 500 3 2)
20 (prestacao 5500 8 4)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
Valor prestação = 1500
Valor prestação = 530
Valor prestação = 7260
```

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)))))))
```

O código em questão define um procedimento adcd, que recebe um parâmetro n e retorna uma lista que contém todos os divisores inteiros de n maiores que 1.

Arquivo 14-operad-logicos.rtk Operadores lógicos

14. Executar o programa e indicar o seu conteúdo

14.1. Escreva e teste pelo menos cinco operações lógicas

```
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  ;; 2023
5  ;; Aluno: Gabriel Costa Fassarella      <===========  seu nome aqui e abaixo
6  ;;
7  #lang racket      ;; define a linguagem default
8  ; -----------------------------------------------
9  (newline)
10 (display "  UENF-CCT-LCMAT-CC, 2023")
11 (newline)
12 (display "  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
13 (newline)
14 (display "  Aluno:  Gabriel Costa Fassarella ")
15 (newline)
16
17 (let ((a 10))
18   (and (>= a 10) (< a 15)))
19
20 (let ((a 8))
21   (or (equal? a 2) (not (equal? a 4))))
22
23 (let ((a 8))
24   (and (equal? a 8) (equal? a 4)))
25
26 (let ((a 8))
27   (or (equal? a 2) (equal? a 4)))
28
29 (display "not(5 > 3) = ") (not(> 5 3))
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2023
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno:  Gabriel Costa Fassarella
#t
#t
#f
#f
not(5 > 3) = #f
```

Arquivo 15-predicados.rtk Predicados

15. Executar o programa e indicar o seu conteúdo

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

```
70
71 ;; Introdução à Linguagem Scheme-Racket
72 ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
73 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
74 ;; 2023
75 ;; Aluno: Gabriel Costa Fassarella      <===========  seu nome aqui e abaixo
76 ;;
77 #lang racket      ;; define a linguagem default
78 ; -----------------------------------------------
79
80 (newline)
81 (display "E' un STRING 'Oi, UENF ...' ?:  ")(string? "Oi, UENF ...")
82 (display "E' un STRING 24 ?:  ")(string? 24)
83 (newline)
84 (display "m e um char? ") (char? 'm)
85 (display "14 e char? ") (char? 14)
86 (display "#\b e um char? ") (char? #\b)
87 (display "#/m e um char? ") (char? #\m)
88
89
90
91
92
```

```
E' un numero REAL 4+5i ?:  #f

E' un numero COMPLEXO 5+3i ?:  #t

E' un STRING 'Oi, UENF ...' ?:  #t
E' un STRING 24 ?:  #f

m e um char? #f
14 e char? #f
#□□□□ e um char? #t
#/m e um char? #t
>
```

Arquivo 16-mapeamentos.rtk Mapeamentos

15.2. Executar o programa e indicar o que ele faz

No código em questão, inicialmente são criadas 3 funções, a primeira calcula o quadrado de um número, a segunda o dobro e a terceira calcula o valor seguinte. Após isso, na linha 39 é criada uma lista e logo em seguida é feito o cálculo da raiz desses números. Novamente é criada uma lista na linha 45 e dessa vez é calculado o quadrado, o dobro e o próximo valor dos números dessa lista.

15.3. Construir um NOVO programa que faça o seguinte mapeamento

$$x \_\rightarrow x^2 + 3x - 9$$

```
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  ;;  2023
5  ;; Aluno: Gabriel Costa Fassarella     <===========  seu nome aqui e abaixo
6  ;;
7  #lang racket      ;; define a linguagem default
8  ; -----------------------------------------------
9
10 (define equac
11   (lambda ( x ) (- (+ (* x x)(* 3 x)) 9)
12   )
13 )
14
15 (define lista (list 1 2 3 4 5))
16
17 (map equac lista)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
'(-5 1 9 19 31)
>
```

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

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

O programa em questão cria inicialmente uma função que recebe 3 parâmetros, sendo esses os 3 índices de um polinômio de segundo grau. A função é responsável por verificar inicialmente se a raiz é degenerada, isso ocorre quando a = 0. Se essa condição não for atendida, é verificada se a raiz é complexa, isso acontece quando delta < 0, e caso isso não ocorrer, as raízes do polinômio são calculados normalmente.

16.1. Fazer testes para outros cinco polinômios de segundo grau

```
74 (newline)
75
76 (display "X^2 + 3X + 5, Raizes = ")
77 (poly2grau 1 3 5)
78 (newline)
79
80 (display "2X^2 + 6X -4, Raizes = ")
81 (poly2grau 2 6 -4)
82 (newline)
83
84 (display "3X -1, Raizes = ")
85 (poly2grau 0 3 -1)
86 (newline)
87
88 (display "X^2 + X + 2, Raizes = ")
89 (poly2grau 1 1 2)
90 (newline)
91
92 (display "X + 7, Raizes = ")
93 (poly2grau 0 1 7)
94 (newline)
```

```
UENF-CCT-LCMAT-CC, 2023
Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
Aluno:  Gabriel Costa Fassarella

X^2 + 3X + 5, Raizes = "Nenhuma Ou Complexa"

2X^2 + 6X -4, Raizes = (list #i0.5615528128088303 #i-3.5615528128088303)

3X -1, Raizes = "degenerada"

X^2 + X + 2, Raizes = "Nenhuma Ou Complexa"

X + 7, Raizes = "degenerada"
```

Arquivo 20-estruturas.rtk Aplicações: Estruturas de dados

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

No código em questão é criado um Struct que apresenta como opções de entrada o nome, o CEP e o telefone do usuário. Em seguida é dada a entrada desses 3 valores e logo em seguida é mostrado na tela.

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

```
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  ;; 2023
5  ;; Aluno: Gabriel Costa Fassarella      <==========  seu nome aqui e abaixo
6  ;;
7    ;; define a linguagem default ===> Habilite Advanced Student
8  ; -----------------------------------------------racket
9
10 (define-struct cadastro (nome idade sexo))
11
12 (define-struct info (email cep fone))
13
14 (define-struct condicao (estado-civil ocupacao))
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student; memory limit: 128 MB.
>
```

Arquivo 30-entrada.rtk

18. Executar o programa e explicar o conteúdo e os resultados
    O código em questão inicia exigindo a entrada de 4 valores de entrada para o usuário, e logo em seguida realiza o cálculo de média e de acordo com o valor do cálculo devolve a situação do usuário (aprovado ou não).
    18.1. Criar um programa NOVO que faça a leitura de dados pessoais de duas pessoas (utilize entrada de dados)

```
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  ;; 2023
5  ;; Aluno: Gabriel Costa Fassarella      <==========  seu nome aqui e abaixo
6  ;;
7  #lang racket      ;; define a linguagem default
8  ; ---------------------------------------------
9
10 (display "Escreva o nome e idade de duas pessoas: ")
11 (newline)
12 (define n1 (read))
13 (define i1 (read))
14 (define n2 (read))
15 (define i2 (read))
16
17 (display "Nome: ") n1
18 (display "Idade: ") i1
19 (display "Nome: ") n2
20 (display "Idade: ") i2
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
Escreva o nome e idade de duas pessoas:
julio
12
ana
25
Nome: 'julio
Idade: 12
Nome: 'ana
Idade: 25
>
```

Parte 2:

Resolver a lista de Exercícios no final da Notas de Aula (Slide 51)

- **Programar em Racket:**
  - **(3 + x)/(7y- 2) – (xy + 9)**

```
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  ;; 2023
5  ;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui
6
7  ;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
8  ;;
9  #lang racket      ;; define a linguagem default
10 ; -----------------------------------------------
11
12 (define eq
13   (lambda (x y)
14     (- (/ (+ 3 x) (- (* 7 y) 2)) (+ (* x y) 9))))
15
16 (eq 2 1)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
-10
>
```

- **Raiz quadrada de $x^2 + 3x - 5$**

```
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2023
;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui

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

(define (bhaskara1 a b c)
  (begin
    (display "x1 = ")
    (let ((delta (-(expt b 2) (* 4 (* a c)))))
        (if (< delta 0)
            (error "delta menor que 0...")
            (/ (- (- b) (sqrt delta)) (* 2 a)))
        )
  )
)

(define (bhaskara2 a b c)
  (begin
    (display "x2 = ")
    (let ((delta (-(expt b 2) (* 4 (* a c)))))
        (if (< delta 0)
            (error "delta menor que 0...")
            (/ (+ (- b) (sqrt delta)) (* 2 a)))
        )
  )
)

(sqrt(bhaskara1 1 5 -3))
(newline)
(sqrt(bhaskara2 1 5 -3))

```

```
Language: racket, with debugging; memory limit: 128 MB.
x1 = 0.0+2.3540138625651951

x2 = 0.7357861544967462
```

- **Criar uma lista com 5 elementos**

```
1  ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
2  ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
3  ;; 2023
4  ;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui
5
6  ;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
7  ;;
8  #lang racket      ;; define a linguagem default
9  ; -----------------------------------------------racket
10
11 (define lista (list 1 2 3 4 5))
12
13 lista
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
'(1 2 3 4 5)
>
```

- **Determinar o segundo elemento de uma lista**

```
1  ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
2  ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
3  ;; 2023
4  ;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui
5
6  ;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
7  ;;
8  #lang racket      ;; define a linguagem default
9  ; -----------------------------------------------racket
10
11 (define lista (list 1 2 3 4 5))
12
13 (display (car (cdr lista)))
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
2
>
```

- **Determinar o antepenúltimo elemento de uma lista dada**

```racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2023
;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui

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

(define lista (list 1 2 3 4 5))

(display (car (cdr (cdr (reverse lista)))))

```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
3
```

- **Consultar se um elemento pertence a uma lista dada**

```racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2023
;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui

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

(define lista (list 1 2 3 4 5))

(define verif
  (lambda (x lista)
    (member? x lista)))

(display (verif 3 lista))
(newline)
(display (verif 6 lista))

```

```
Welcome to DrRacket, version 8.8 [cs].
Language: Advanced Student [custom]; memory limit: 128 MB.
Teachpack: gui.rkt.
#true
#false
>
```

- **Adicionar o terceiro elemento de uma lista**
  - **No final de outra lista A**
  - **No início de outra lista B**

```
1  ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
2  ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
3  ;; 2023
4  ;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui
5
6  ;;;;;;;;;;;;;;;;   Escolha a linguagem "Determine language from source"
7  ;;
8  #lang racket      ;; define a linguagem default
9  ; -----------------------------------------------racket
10 (define listaA (list 1 2 3 4 5))
11 (define listaB (list 6 7 8 9 10))
12
13 (display (append listaA (list (car (cdr (cdr listaB))))))
14
15 (display (cons (list (car (cdr (cdr listaA)))) listaB))
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
(1 2 3 4 5 8)((3) 6 7 8 9 10)
>
```

- **Calcular o perímetro de um quadrado, circulo ou triangulo**

```racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2023
;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui

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

(define quad (lambda (l)
      (* l 4)
    )
  )

(define circ (lambda (r)
      (* (* 2 3.14) r)
    )
  )

(define tri (lambda (l1 l2 l3)
      (+ (+ l1 l2) l3)
    )
  )

(quad 2)
(circ 5)
(tri 3 5 4)
```

```
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
8
31.400000000000002
12
>
```

- **Calcular o k-ésimo número inteiro (par ou ímpar)**

```racket
;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2023
;; Aluno: Gabriel Costa Fassarella      <==========  escreva seu nome aqui

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

(define (k-enesimo k)
  (if (even? k)
      (* (/ k 2) 1)
      (if (= k 1)
          1
          (* (ceiling (/ k 2)) -1))))

(k-enesimo 10)
```