



Disciplina: **AARE Paradigmas de Linguagens de Programação**

Professor: Prof. Ausberto S. Castro V.

E-mail: ascv@uenf.br

Data: 20 de abril de 2024

Prática – Racket

Nome Completo: Lucas Nathã Monteiro Pimentel de Souza

Data: 20 de abril de 2024

Total Exercícios Resolvidos:

Arquivo 01-primeiro.rkt **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?

```
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 ;; 2024
5 ;; Aluno: Fulano <===== 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, Nota 5 MEC 2024")
13   (newline)
14   (display "Bemvindo a Linguagem Racket-Scheme! ") ;<----- mostrar na tela
15   (newline)
16   (display "Sou o aluno Fulano de Tal")
17 )
```

Codigo fonte

Interações (shell)

Explicar aqui:

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  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza
6  ;;
7  ;; Linguagem Advanced Student          ;; define a linguagem default
8  ;; O primeiro programa Racket
9  ;; -----
10 (begin
11   (newline) ; nova linha
12   (display "Bom dia, UENF 30 anos, Nota 5 MEC 2024") ; imprimir na tela a msg entre aspas
13   (newline) ; nova linha
14   (display "Bem vindo a Linguagem Racket-Scheme! ") ;<----- mostrar na tela
15   (newline) ; nova linha
16   (display "Sou o aluno Lucas Nathã Monteiro Pimentel de Souza") ; imprimir na tela a msg entre aspas
17   (newline)
18   (display "Prática 01 - Linguagem Racket, Lucas Nathã Monteiro Pimentel de Souza, 18/04/2024")
19 )
20
21

```

Welcome to [DrRacket](#), version 8.12 [cs].

Language: [Advanced Student](#); memory limit: 128 MB.

Bom dia, UENF 30 anos, Nota 5 MEC 2024

Bem vindo a Linguagem Racket-Scheme!

Sou o aluno Lucas Nathã Monteiro Pimentel de Souza

Prática 01 - Linguagem Racket, Lucas Nathã Monteiro Pimentel de Souza, 18/04/2024

>

Arquivo [02-numeros.rtk](#) [Números e Aritmética](#)

3. Execute o programa e mostre os resultados

```

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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6 ;;
7 ;;;;;;;;;;;;;; Escolha a linguagem "Determine language from source"
8 ;;
9
10 ; -----

```

Welcome to [DrRacket](#), version 8.12 [cs].

Language: [Advanced Student](#); memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)

Aluno: Lucas Nathã Monteiro Pimentel de Souza

Soma $13 + 18 = 31$

Produto $57 * 92 = 5244$

Mistura $7 + (3 * 9) = 34$

Combinando $(2 + (3 * 4)) / 2 - 4 = 3$

Raiz quadrada de 6 = #i2.449489742783178

Raiz quadrada de 14= #i3.7416573867739413

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 $6^3 = 216$

exponente $4^{(1/2)} = 2$

Maximo de 3 32 42 27 3 = 42

minimo de 31 3 42 25 9 = 3

valor absoluto de 18 = 18

valor absoluto de -7 = 7

"quotes obriga as listas serem tratadas como DADOS"

(list 8 1 9)

(list '/ 4 (list '* 3 7))

(list 1 'a 9 'b)

(list (list 'a 'b) (list 9 5))

(list '+ 7 (list '* 5 4))

4. Escreva programas Racket para as seguintes expressões:

4.1. $P = (4 - (4^2 - 7^3) / 8) - (11 + (5 - (3^4 - 1)))$

```

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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6 ;;
7 ;;:::::::::::::::::::: Escolha a linguagem "Determine language from source"
8 ;;
9
10 ; -----
11 (display "    UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display "    Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display "    Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 (newline)
17 ;;Exercicios pedidos na lista
18 (newline)
19 (newline)
20 (display "Questão 4.1")
21 (newline)
22 (- (- 4 (/ (- (expt 4 2) (expt 7 3)) 8)) (+ 11 (- 5 (- 34 1))))
23
24
25
26
27
28
29
30
31

```

Welcome to [DrRacket](#), version 8.12 [cs].

Language: *Advanced Student*; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)

Aluno: Lucas Nathã Monteiro Pimentel de Souza

Questão 4.1

61.875

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

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

```

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  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6  ;;
7  ;;:::::::::::::::::::: Escolha a linguagem "Determine language from source"
8  ;;
9
10 ; -----
11 (display "      UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display "      Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display "      Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 (newline)
17 ;;Exercicios pedidos na lista
18 (newline)
19 (display "Questão 4.2")
20 (newline)
21 (/ (+ (sqrt (+ (+ (+ (expt 5 2) 6) (sin (- 12 8))) (cos (+ 20 25))))) (* (- 5 3) (expt (+ 4 8) 2)))
22
23
24
25
26
27
28
29
30
31

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: Advanced Student; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno: Lucas Nathã Monteiro Pimentel de Souza

Questão 4.2
#i0.01926020086234269

```

Arquivo [03-variaveis.rtk](#)

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

```

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  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6
7  ;;:::::::::::::::::::: Escolha a linguagem "Determine language from source"
8  ;;
9  #lang racket      ;; define a linguagem default
10 ; -----
11 (display "  UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display "  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display "  Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 (newline)
17 ;;
18 ;;
19 ;; Variáveis e Expressões LET
20 ;; -----
21 (define y 5)                ; para y = 5
22 (define m 7)                ; para m = 7
23 (define z (+ m 6))          ; para z = m + 6      z = 7 + 6 => z = 13
24 (define Pi 3.141516)
25 (define k (+ z (- m y)))    ; para k = z + (m - y) => k = 13 + (7 - 5) => k = 15
26
27 (display "m = ")
28 m
29 (display "z = ")
30 z
31 (display "k = ")
32 k
33
34

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno: Lucas Nathã Monteiro Pimentel de Souza
m = 7
z = 13
k = 13

```

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

```

13 (display " Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display " Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 (newline)
17 ; Expressao let: (let ((var1 valor) ...) expr1 expr2 ..... )
18 (display "let é definir o valor de uma variável")
19 (newline)
20 (newline)
21 (display "x = 24, x + 6 = 30")
22 (newline)
23 (let ((x 24)) ;x = 24
24   (+ x 6)) ;24 + 6 = 30
25
26 ;;:
27 (newline)
28 (display "a = 5 e b = 8, 3 + (b * a) = 43")
29 (newline)
30 (let ( (a 5) (b 8)) ;a = 5 e b = 8
31   (+ 3 (* a b))) ; 3 + (8 * 5) = 43
32
33 ;;:
34 (newline)
35 (display "op1 = + e op2 = * e x = 5, 3 + (4 * x) = 23")
36 (newline)
37 (let ( (op1 +) ;op1 = +
38       (op2 *) ;op2 = *
39       (x 5) ;x = 5
40     )
41   (op1 3 (op2 4 x)) ; 3 + (4 * 5)
42 )

```

```

UENF-CCT-LCMAT-CC, 2024
Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
Aluno: Lucas Nathã Monteiro Pimentel de Souza
let é definir o valor de uma variável

x = 24, x + 6 = 30
30

a = 5 e b = 8, 3 + (b * a) = 43
43

op1 = + e op2 = * e x = 5, 3 + (4 * x) = 23
23

```

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

```

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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6
7 :::::::::::::::::::: Escolha a linguagem "Determine language from source"
8 ;;
9 #lang racket      ;; define a linguagem default
10 ; -----
11 (display "  UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display "  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display "  Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 (newline)
17 ; Expressoes pedidas no exercicio 5.2
18 (let ((p 2)) ; p = 2
19     (* p 30) ; 2 * 30 = 60
20 )
21
22 (let ((l 22)) ; l = 22
23     (+ 3 (/ l 2))) ; 3 + (22 / 2) = 14
24
25
26
27
28
29
30
31
32
33
34
35

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno: Lucas Nathã Monteiro Pimentel de Souza
60
14

```

Arquivo [04-areas.rtk](#)

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


```

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  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza    <===== escreva seu nome aqui
6
7  ;;;;;;;;;;;;;;; Escolha a linguagem "Determine language from source"
8  ;;
9  #lang racket      ;; define a linguagem default
10 -----
11 (display "  UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display "  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display "  Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 ;;
17 ;;
18 ;; (define (nomeFuncao parametros) definicao )
19 ;;
20 ;; AREAS
21 ;; -----
22 (define (area-circulo r) ;função para calcular a área do circulo
23   (* 3.14 (* r r))      ) ; 3.14 * (r * r)
24
25 (define (areatriangulo b h) ;função para calcular a área do triângulo
26   (/ (* b h) 2)) ;função para calcular a area do triângulo
27
28 (define (areadisco interno externo) ;função para calcular a área do disco
29   (- (area-circulo externo)
30      (area-circulo interno) ;chama a função area-circulo para saber o valor do circulo externo e interno e dps faz externo - interno
31   )
32 )
33

```

Language: racket, with debugging; memory limit: 128 MB.

```

  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno: Lucas Nathã Monteiro Pimentel de SouzaCirculo de raio 21  AREA = 1384.74
  Triangulo base=10, altura=15  AREA = 75
  Disco raio menor=12, raio maior=20  AREA = 803.8399999999999

```

0 programa calcula a área do circulo, triangulo e de um disco

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.

```

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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza    <===== escreva seu nome aqui
6
7 ;;;;;;;;;;;;;; Escolha a linguagem "Determine language from source"
8 ;;
9 #lang racket      ;; define a linguagem default
10 ; -----
11 (display " UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display " Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display " Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 (newline)
17 (newline)
18 ;;Exercicio pedido na lista
19 (define (area-quadrado l)
20   (* l l)
21 )
22
23 (display "Quadrado de lado 2, AREA = ")
24 (area-quadrado 2)
25
26 (define (area-trapezio b1 b2 h)
27   (* (/ h 2) (+ b1 b2))
28 )
29
30 (display "Trapézio b1 = 2, b2 = 4 e h = 3, AREA = ")
31 (area-trapezio 2 4 3)
32

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno: Lucas Nathã Monteiro Pimentel de Souza

Quadrado de lado 2, AREA = 4
Trapézio b1 = 2, b2 = 4 e h = 3, AREA = 9

```

- 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 Scheme-Racket
2  ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3  ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza    <===== escreva seu nome aqui
6
7  ;;:::::::::::::::::::: Escolha a linguagem "Determine language from source"
8  ;;
9  #lang racket      ;; define a linguagem default
10 ; -----
11 (display "  UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display "  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display "  Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 (newline)
17 (newline)
18 ;;Exercicio pedido na lista
19 (define (area-galao R A)
20   (* 3.14 (* (* R R) A))
21 )
22
23 (display "Galão com R = 3 e A = 6, Volume = ")
24 (area-galao 3 6)
25
26
27
28
29
30
31

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno: Lucas Nathã Monteiro Pimentel de Souza

Galão com R = 3 e A = 6, Volume = 169.56

```

Arquivo [05-funcoes.rtk](#)

7. Execute o programa e explique o que faz

```

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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6
7 ;;:::::::::::::::::::: Escolha a linguagem R5RS
8 ;;
9 #lang racket      ;; define a linguagem default
10 ; -----
11 (display "  UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display "  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display "  Aluno: Lucas Nathã Monteiro Pimentel de Souza ")
16 (newline)
17 ;;
18 ;;
19 ;; -----
20
21 ;;-----
22 ;; Procedimento (lambda ParametrosFormais Corpo)
23 (define ADICIONA
24   (lambda (a b)      ; parametros formais
25     (+ a b ) ; a + b      ; corpo do procedimento ADICIONA
26   )

```

```

  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno: Lucas Nathã Monteiro Pimentel de Souza

```

```

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 cria funções/procedimentos utilizando o lambda

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

```

10 ; -----
11 (display " UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display " Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display " Aluno: Lucas Nathã Monteiro Pimentel de Souza ")
16 (newline)
17 ;Exercicios da lista
18 (define cubo
19   (lambda (x)
20     (begin
21       (newline)
22       (display "O cubo de ")
23       (display x)
24       (display " é = ")
25       (expt x 3)
26     )))
27
28 (define raizquadrada
29   (lambda (x)
30     (begin
31       (newline)
32       (display "A raiz quadrada de ")
33       (display x)
34       (display " é = ")
35       (sqrt x)
36     )))
37 (cubo 5)
38 (raizquadrada 25)
39
40 (newline)
41 ..

```

Welcome to [DrRacket](#), version 8.12 [cs].

Language: racket, with debugging; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)

Aluno: Lucas Nathã Monteiro Pimentel de Souza

O cubo de 5 é = 125

A raiz quadrada de 25 é = 5

Arquivo [06-condicionalIF.rtk](#)

8. Executar e explicar o programa

```

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  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6
7  ;;:::::::::::::::: Escolha a linguagem R5RS
8  ;;
9  ;; Ajuda: http://docs.racket-lang.org/guide/syntax-overview.html#(part._Conditionals_with_if_and_or_and_cond)
10 ;;
11 #lang racket      ;; define a linguagem default
12 ; -----
13 (display " UENF-CCT-LCMAT-CC, 2024")
14 (newline)
15 (display " Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
16 (newline)
17 (display " Aluno: Lucas Nathã Monteiro Pimentel de Souza")
18 (newline)
19
20 ;Exercicios da lista
21
22 (define n1 3)
23 (define n2 8)
24 (define m (/ (+ n1 n2) 2))
25
26 (if (>= m 6)
27     (display "Aprovado!")
28     (display "Reprovado!"))
29 \

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno: Lucas Nathã Monteiro Pimentel de Souza
Reprovado!
R diferente de 10
P = 528
P maior que 57
MinQadrado de 6 e 9 = 36

Oi pessoal da Computacao UENF! "Tudo Bem?"
Vai chover na madrugada? "Nao entendi."
Funções/procedimentos utilizando o condicional IF

```

8.1. Escreva um programa com dois condicionais

```

4 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6
7 ;;;;;;;;;;;;;; Escolha a linguagem R5RS
8 ;;
9 ;; Ajuda: http://docs.racket-lang.org/guide/syntax-overview.html#(part._Conditionals_with_if_and_or_and_cond
10 ;;
11 #lang racket ;; define a linguagem default
12 ; -----
13 (display " UENF-CCT-LCMAT-CC, 2024")
14 (newline)
15 (display " Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
16 (newline)
17 (display " Aluno: Lucas Nathã Monteiro Pimentel de Souza")
18 (newline)
19
20 ;Exercicios da lista
21
22 (define R 10)
23 (if (= R 10.3)
24     (display "R igual 10")
25     (display "R diferente de 10"))
26 )
27
28 (newline)

```

Welcome to [DrRacket](#), version 8.12 [cs].
 Language: racket, with debugging; memory limit: 128 MB.
 UENF-CCT-LCMAT-CC, 2024
 Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
 Aluno: Lucas Nathã Monteiro Pimentel de Souza
 R diferente de 10

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

```

7 ;;;;;;;;;;;;;; Escolha a linguagem R5RS
8 ;;
9 ;; Ajuda: http://docs.racket-lang.org/guide/syntax-overview.html#(part._Conditionals_with_if_and_or_and_cond)
10 ;;
11 #lang racket ;; define a linguagem default
12 ; -----
13 (display " UENF-CCT-LCMAT-CC, 2024")
14 (newline)
15 (display " Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
16 (newline)
17 (display " Aluno: Lucas Nathã Monteiro Pimentel de Souza")
18 (newline)
19
20 ;Exercicios da lista
21
22 (define n1 3)
23 (define n2 8)
24 (define m (/ (+ n1 n2) 2))
25
26 (if (>= m 6)
27     (display "Aprovado!")
28     (display "Reprovado!"))
29 )
30
31

```

Welcome to [DrRacket](#), version 8.12 [cs].
 Language: racket, with debugging; memory limit: 128 MB.
 UENF-CCT-LCMAT-CC, 2024
 Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
 Aluno: Lucas Nathã Monteiro Pimentel de Souza
 Reprovado!

9. Executar e explicar o programa

```
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 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (display " UENF-CCT-LCMAT-CC, 2024")
10 (newline)
11 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display " Aluno: Lucas Nathã Monteiro Pimentel de Souza")
14 (newline)
15
16 ;;Exercicio da lista
17
18 (define (fac x)
19   (let loop ((x x) (acc 1))
20     (if (zero? x)
21         acc
22         (loop (sub1 x) (* x acc)))))
23
24 (newline)
```

Language: racket, with debugging; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)

Aluno: Lucas Nathã Monteiro Pimentel de Souza

Factorial de 5 (segunda forma) = 120

Pitagoras de 3 e 4 = 25

Aproximando Pi com 4 = $3\frac{107}{315}$

Aproximando Pi com 10 = $3\frac{3380087}{14549535}$

Factorial de 5 = 120

Testando algumas formulas conhecidas como fatorial, pitagoras e aproximação de pi

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


```

4 ;; Abril 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----|
9 (display " UENF-CCT-LCMAT-CC, 2024")
10 (newline)
11 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display " Aluno: Lucas Nathã Monteiro Pimentel de Souza")
14 (newline)
15
16 ;;Exercicio da lista
17
18 (define (fac x)
19   (let loop ((x x) (acc 1))
20     (if (zero? x)
21         acc
22         (loop (sub1 x) (* x acc)))))
23
24 (newline)
25 (display "Factorial de 5 (segunda forma) = ")
26 (fac 5)
27

```

```

Factorial de 5 (segunda forma) = 120

```

Arquivo [08-condicional.rtk](#)

10. Execute o programa e indique o que faz


```

16 |
17 | (define (computador preco)
18 |   (cond
19 |     ((<= preco 800) "Muito ruim")
20 |     ((<= preco 1500) "Ruim")
21 |     ((<= preco 3000) "Mediano")
22 |     ((<= preco 4500) "Bom")
23 |     ((<= preco 7000) "Muito bom")
24 |     ((<= preco 10000) "Excelente")
25 |     (else "Necessário apenas para uso específico")
26 |   )
27 | )
28 |
29 |
30 | (newline)
31 | (display "Computador de R$800,00 = ")
32 | (computador 800)
33 |
34 | (newline)
35 | (display "Computador de R$1.500,00 = ")
36 | (computador 1500)
37 |
38 | (newline)
39 | (display "Computador de R$3.000,00 = ")
40 | (computador 3000)
41 |

```

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)

Aluno: Lucas Mathã Monteiro Pimentel de Souza

Computador de R\$800,00 = "Muito ruim"

Computador de R\$1.500,00 = "Ruim"

Computador de R\$3.000,00 = "Mediano"

Computador de R\$4.500,00 = "Bom"

Computador de R\$7.000,00 = "Muito bom"

Computador de R\$10.000,00 = "Excelente"

Computador de R\$15.000,00 = "Necessário apenas para uso específico"

- 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

```

59 |
60 |
61 | (define (delta a b c)
62 |   (+ (expt b 2) (* (* -4 a) c))
63 | )
64 |
65 | (define (x1 a b c delta)
66 |   (display "x1 = ")
67 |   ;(delta a b c)
68 |   (/ (+ b (sqrt delta)) (* 2 a))
69 | )
70 |
71 | (define (x2 a b c delta)
72 |   (display "x2 = ") (/ (- b (sqrt delta)) (* 2 a))
73 | )
74 |
75 | (newline)
76 | (display "a = 25, b = -55, c = 10")
77 | (newline)
78 | (display "delta = ")
79 | (delta 25 55 10)
80 | (x1 25 55 10 (delta 25 55 10))
81 | (x2 25 55 10 (delta 25 55 10))
82 |
83 | (newline)
84 | (newline)
85 | ;;
86 | ;;
87 | ;; CONDICIONAL ( cond {[ <expr-test> <expr-eval>* ]}* )
88 | ;;-----
--

```

```

a = 25, b = -55, c = 10
delta = 2025
x1 = 2
x2 =  $\frac{1}{5}$ 

```

Arquivo [09-pares.rtk](#)

11. Execute o programa

```

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 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (newline)
10 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2024")
11 (newline)
12 (display " Aluno: Lucas Nathã Monteiro Pimentel de Souza")
13 (newline)
14
15 ;;
16 ;;
17 ;; PARES CONSTANTES
18 ;; -----
19
20 (define x (cons 7 9)) ;pares constantes da variavel x é 7 e 9
21 (define y (cons 'a 4)) ;pares constantes da variavel y é 'a e 4
22 (define z (cons x y)) ; ( (7 9) ('a 4) ) pares constantes da variavel z é (7 9) e ('a 4)
23
24 (define a (cons 5 6)) ;pares constantes da variavel a é 5 e 6

```

```

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2024
Aluno: Lucas Nathã Monteiro Pimentel de Souza

```

```

0 par x = '(7 . 9)

0 par y = '(a . 4)

0 par z = (x y) = '((7 . 9) a . 4)

0 par a4 = '(((5 . 6) . 6) . 4) . 8)

0 par b4 = '(8 4 6 6 . 5)

0 primeiro elemento do par x = 7

0 segundo elemento do par x = 9
>

```

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 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (newline)
10 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2024")
11 (newline)
12 (display " Aluno: Lucas Nathã Monteiro Pimentel de Souza")
13 (newline)
14 ;;EXERCICIOS DA LISTA
15 (display "Exercicios da lista")
16 (newline)
17
18 (define k (cons 72 93))
19 (define j (cons 2 11))
20
21 (newline)
22 (display "0 par k = ")
23 k
24
25 (newline)

```

```

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2024
Aluno: Lucas Nathã Monteiro Pimentel de Souza
Exercicios da lista

0 par k = '(72 . 93)

0 par j = '(2 . 11)

0 primeiro elemento do par k = 72

0 segundo elemento do par k = 93

0 primeiro elemento do par j = 2

0 segundo elemento do par j = 11

```

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  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6  ;;
7  #lang racket      ;; define a linguagem default
8  ;; define a linguagem default: R5RS
9  ; -----
10 (display " UENF-CCT-LCMAT-CC, 2024")
11 (newline)
12 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)"
13 (newline)
14 (display " Aluno: Lucas Nathã Monteiro Pimentel de Souza")
15 (newline)
16 ;Exercicio da lista 12.1
17
18 (define list '(2 3 1 4 10 11))
19 (define first (car list))
20 (define ultimo (last list))
21 (define comprimento (length list))
22 (define newlist (append list '(9 5)))
23
24 (display "Lista = ")
25 list
26 (display "Primeiro elemento = ")
27 first
28 (display "Ultimo elemento = ")
29 ultimo
30 (display "Comprimento = ")
31 comprimento
32 (display "Nova lista = ")
33 newlist

```

Welcome to [DrRacket](#), version 8.12 [cs].

Language: racket, with debugging; memory limit: 128 MB.

```

UENF-CCT-LCMAT-CC, 2024
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
Aluno: Lucas Nathã Monteiro Pimentel de Souza
Lista = '(2 3 1 4 10 11)
Primeiro elemento = 2
Ultimo elemento = 11
Comprimento = 6
Nova lista = '(2 3 1 4 10 11 9 5)

```

- 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)

```

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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6 ;;
7 #lang racket      ;; define a linguagem default
8 ;; define a linguagem default: R5RS
9 ; -----
10 (display "  UENF-CCT-LCMAT-CC, 2024")
11 (newline)
12 (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
13 (newline)
14 (display "  Aluno: Lucas Nathã Monteiro Pimentel de Souza")
15 (newline)
16 ;Exercicio da lista 12.2
17
18 (define llist (append '(4 7 2) '(9 8 7) '(1 6 2) '(3 4)))
19 (newline)
20 (display llist)
21 (newline)
22
23
24
25
26
27
28
29
30
31
32
33

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno: Lucas Nathã Monteiro Pimentel de Souza

(4 7 2 9 8 7 1 6 2 3 4)

```

Arquivo [12-lambda.rtk](#) e [13-lambda.rtk](#)

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


```

51 (display "3+3 = ") ;faz a mesma operação 2x (nesse caso passamos a soma)
52 (DuasVezes + 3)
53
54 (display "3*3 = ") ;faz a mesma operação 2x (nesse caso passamos a multiplicação)
55 (DuasVezes * 3)
56
57 (display "3-3 = ") ;faz a mesma operação 2x (nesse caso passamos a subtração)
58 (DuasVezes - 3)
59
60 (newline)
61 (display "8+5 = ") ;função q utiliza sempre o 5 e faz alguma operação de acordo com oque passamos (nesse caso soma)
62 (com5 + 8)
63
64 (display "8*5 = ") ;função q utiliza sempre o 5 e faz alguma operação de acordo com oque passamos (nesse caso multiplicação)
65 (com5 * 8)
66
67 (display "8-5 = ") ;função q utiliza sempre o 5 e faz alguma operação de acordo com oque passamos (nesse caso subtração)
68 (com5 - 8)
69 (newline)
70
71 (display "P(x) = X^2 + 3X - 7   entao P(7) = " ) ;calcula o polinômio utilizando x = 7
72 (polinomio 7)
73
74 (display "P(x) = X^2 + 3X - 7   entao P(1) = " ) ;calcula o polinômio utilizando x = 1
75 (polinomio 1)
76
77 (display "P(x) = X^2 + 3X - 7   entao P(3) = " ) ;calcula o polinômio utilizando x = 3
78 (polinomio 3)
79

```

```

UENF-CCT-LCMAT-CC, 2024
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
Sou o aluno(a): Lucas Nathã Monteiro Pimentel de Souza

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(7) = 63
P(x) = X^2 + 3X - 7   entao P(1) = -3
P(x) = X^2 + 3X - 7   entao P(3) = 11

```

13.1. Crie um procedimento para realizar o cálculo de uma prestação em atraso, utilizando a fórmula $\text{Prest} = \text{valor} + (\text{valor} * (\text{taxa}/100) * \text{tempo})$. 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  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== seu nome aqui e abaixo
6  ;;
7  #lang racket      ;; define a linguagem default
8  ; -----
9  (newline)
10 (display "UENF-CCT-LCMAT-CC, 2024")
11 (newline)
12 (display "Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
13 (newline)
14 (display "Sou o aluno(a): Lucas Nathã Monteiro Pimentel de Souza")
15 (newline)
16 ;;Exercicio da lista 13.1
17 (newline)
18 (define (prestacao val taxa tempo)
19   (+ val (* (/ (* val taxa) 100) tempo))
20 )
21
22 (display "valor = 1000, taxa = 10%, tempo = 2 meses => ")
23 (prestacao 1000 10 2)
24
25 (display "valor = 1700, taxa = 35%, tempo = 8 meses => ")
26 (prestacao 1700 35 8)
27
28
29
30
31
32
33 ..

```

Welcome to [DrRacket](#), version 8.12 [cs].

Language: racket, with debugging; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)

Sou o aluno(a): Lucas Nathã Monteiro Pimentel de Souza

valor = 1000, taxa = 10%, tempo = 2 meses => 1200

valor = 1700, taxa = 35%, tempo = 8 meses => 6460

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)))))))

```

```

1  ;; Introdução à Linguagem Scheme-Racket
2  ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
3  ;; UENF-CCT-LCHAT - Curso de Ciencia da Computacao
4  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== seu nome aqui e abaixo
6  ;;
7  #lang racket      ;; define a linguagem default
8  ; -----
9  (newline)
10 (display "UENF-CCT-LCHAT-CC, 2024")
11 (newline)
12 (display "Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
13 (newline)
14 (display "Sou o aluno(a): Lucas Nathã Monteiro Pimentel de Souza")
15 (newline)
16 ;Exercicio da lista 13.2
17 (define abcd
18   (lambda (n)
19     (let f ((i 2))
20       (cond
21         ((>= i n) '())
22         ((integer? (/ n i))
23          (cons i (f (+ i 1))))
24         (else (f (+ i 1)))))))
25
26 (newline)
27 (newline)
28 (display "define uma função lambda e define uma função f com parametro i = 2, e logo em seguida chama uma condicional testando se i >= n, se isso for verdade retorna uma lista vazia")
29 (newline)
30 (display "se a condição i >= n for falsa ele verifica se n dividido por i da resto 0 e caso não dê, ele chama a função f dnv adicionando +1 na variável i")
31
32
33

```

Welcome to [DrRacket](#), version 8.12 [cs].
Language: racket, with debugging; memory limit 128 MB.

UENF-CCT-LCHAT-CC, 2024
Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
Sou o aluno(a): Lucas Nathã Monteiro Pimentel de Souza

define uma função lambda e define uma função f com parametro i = 2, e logo em seguida chama uma condicional testando se i >= n, se isso for verdade retorna uma lista vazia
se a condição i >= n for falsa ele verifica se n dividido por i da resto 0 e caso não dê, ele chama a função f dnv adicionando +1 na variável i

Arquivo [14-operad-logicos.rtk](#) Operadores lógicos
14. Executar o programa e indicar o seu conteúdo

```

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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== seu nome aqui e abaixo
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (newline)
10 (display "UENF-CCT-LCMAT-CC, 2024")
11 (newline)
12 (display "Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
13 (newline)
14 (display "Sou o aluno(a): Lucas Nathã Monteiro Pimentel de Souza")
15 (newline)
16 ;;
17 ;;
18 (display "Verificação com operadores lógicos e tabela verdade") ;;RESPOSTA DA 14
19
20
21 (newline)
22 (newline)
23 (display "Operadores lógicos... ")
24 (newline)
25
26 (let ((x 3))
27   (and (> x 2) (< x 4)))
28
29 (let ((y 5))
30   (and (> y 2) (< y 4)))
31

```

Welcome to [DrRacket](#), version 8.12 [cs].
 Language: racket, with debugging; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2024
 Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
 Sou o aluno(a): Lucas Nathã Monteiro Pimentel de Souza
 Verificação com operadores lógicos e tabela verdade

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  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== seu nome aqui e abaixo
6  ;;
7  #lang racket      ;; define a linguagem default
8  ; -----
9  (newline)
10 (display "UENF-CCT-LCMAT-CC, 2024")
11 (newline)
12 (display "Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
13 (newline)
14 (display "Sou o aluno(a): Lucas Nathã Monteiro Pimentel de Souza")
15 (newline)
16 ;;
17 ;;
18 ;Exercicio da lista 14.1
19
20 (display "510 > 333? ") (> 5 3)
21 (display "510 = 510.1? ") (= 510 510.1)
22 (display "500 = 500? ") (= 500 500)
23 (display "not(500 = 500)? ") (not (= 500 500))
24 (display "not(500 > 600)? ") (not (> 500 600))
25
26
27
28
29
30
31
32

```

Welcome to [DrRacket](#), version 8.12 [cs].

Language: racket, with debugging; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)

Sou o aluno(a): Lucas Nathã Monteiro Pimentel de Souza

510 > 333? #t

510 = 510.1? #f

500 = 500? #t

not(500 = 500)? #f

not(500 > 600)? #t

Arquivo [15-predicados.rtk](#) Predicados

15. Executar o programa e indicar o seu conteúdo

```

52 |
53 | (display "Numero 6.78 ?:" ) (number? 6.78) ;função que verifica se o que foi digitado é um número
54 | (display "Numero 2/7 ?:" ) (number? 2)
55 | ;
56 | (newline)
57 | (display "E' un numero INTEIRO 3 ?:" ) (integer? 3) ;função integer verifica se um número é inteiro ou não (voltando #t se for inteiro e #f se não for)
58 | (display "E' un numero INTEIRO 3.0 ?:" ) (integer? 3.0)
59 | (display "E' un numero INTEIRO 3/7 ?:" ) (integer? 3/7)
60 | ;;
61 |
62 | (newline)
63 | (display "E' un numero REAL 2 ?:" ) (real? 2) ;função real verifica se um número é real ou não (exemplos de números reais: 3.5 3/2 7/3)
64 | (display "E' un numero REAL 3/5 ?:" ) (real? 3/5)
65 | (display "E' un numero REAL 4+5i ?:" ) (real? 4+5i)
66 | ;
67 | ;;
68 | (newline)
69 | (display "E' un numero COMPLEXO 5+3i ?:" ) (complex? 5+3i) ;função complex verifica se um número é complexo ou não (números complexos contem a variavel i q tem o valor de -1)
70 | ;;
71 |
72 | (newline)
73 | (display "E' un STRING 'Oi, UENF ...' ?:" ) (string? "Oi, UENF ...") ;função que verifica se o que foi digitado é uma string
74 | (display "E' un STRING 24 ?:" ) (string? 24)
75 | ;
76 |
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 |

```

Welcome to [DrRacket](#), version 8.12 [cs].
 Language: racket, with debugging, memory limit 128 MB.
 UENF-CCT-LCHAT-CCT, 2024
 Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
 Aluno(a): Lucas Mathã Nonteiro Pimentel de Souza
 Operador IGUAL
 a = 7? #f
 4 = 4? #t
 car = car? #t
 car = cdr? #f

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  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== seu nome aqui e abaixo
6  ;;
7  #lang racket      ;; define a linguagem default
8  ; -----
9  (display "UENF-CCT-LCMAT-CC, 2024")
10 (newline)
11 (display "Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
12 (newline)
13 (display "Aluno(a): Lucas Nathã Monteiro Pimentel de Souza")
14 (newline)
15 ;;Exercicio da lista 15.1
16 (display (char? 'm)) (newline)
17 (display (char? 14)) (newline)
18 (display (char? #\b)) (newline)
19 (display (char? #\m)) (newline)
20
21
22
23
24
25
26
27
28
29
30
31
32
33

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: racket, with debugging; memory limit: 128 MB.
UENF-CCT-LCMAT-CC, 2024
Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
Aluno(a): Lucas Nathã Monteiro Pimentel de Souza
#f
#f
#t
#t

```

Arquivo [16-mapeamentos.rtk](#) Mapeamentos

15.2. Executar o programa e indicar o que ele faz

```

29 )
30 ;;;;
31 (define proximo
32   (lambda ( x ) (+ x 1)
33   )
34 )
35
36 ;;
37 ;;
38
39 (define Lista (list 9 49 81 37 83))
40 (display "Lista: ") Lista
41 (display "Raiz Lista: ") (map sqrt Lista) ;faz a raiz quadrada de todos os elementos da lista
42 ;;
43 ;;
44 (newline)
45 (define dados '(11 6 35 51))
46 (newline)
47 (display "Dados: ") dados
48 (newline)
49 ;;mapeamentos
50 (display "Mapeamentos: Dados-Proximo ") (newline)
51 (map proximo dados) ;vai pegar a lista dados e todos os elementos que tem nela serão somados +1
52
53 (display "Mapeamentos: Dados-Quadrado ") (newline)
54 (map quadrado dados) ;vai pegar a lista dados e todos os elementos que tem nela serão elevados ao quadrado
55
56 (display "Mapeamentos: Dados-Dobro ") (newline)
57 (map dobro dados) ;vai pegar a lista dados e todos os elementos que tem nela serão multiplicados por 2
58
59
60

```

Welcome to [DrRacket](#), version 8.12 [cs].

Language: racket, with debugging; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)

Aluno(a): Lucas Nathã Monteiro Pimentel de Souza

Lista: '(9 49 81 37 83)

Raiz Lista: '(3 7 9 6.082762530298219 9.1104335791443)

15.3. Construir um NOVO programa que faça o seguinte mapeamento

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


```

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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== seu nome aqui e abaixo
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (newline)
10 (display "UENF-CCT-LCMAT-CC, 2024")
11 (newline)
12 (display "Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
13 (newline)
14 (display "Aluno(a): Lucas Nathã Monteiro Pimentel de Souza")
15 (newline)
16 ;;Exercicio da lista 15.3
17 (define (map1 a)
18   (- (+ (* a a) (* 3 a)) 9)
19 )
20 (newline)
21 (display "Mapeamento de 20 = ") (map1 20)
22 (display "Mapeamento de 30 = ") (map1 30)
23 (display "Mapeamento de 40 = ") (map1 40)
24 (display "Mapeamento de 50 = ") (map1 50)
25 (newline)
26
27
28
29
30
31

```

Welcome to [DrRacket](#), version 8.12 [cs].

Language: racket, with debugging; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)

Aluno(a): Lucas Nathã Monteiro Pimentel de Souza

Mapeamento de 20 = 451

Mapeamento de 30 = 981

Mapeamento de 40 = 1711

Mapeamento de 50 = 2641

Arquivo [17-raizes-poly.rtk](#) Aplicações: Raízes de polinômios
 16. Executar o programa e explicar o conteúdo e os resultados

```

40 | (newline)
41 | ;; EXEMPLOS
42 | (display "X^2 + 2X + 1 = 0 , Raizes = ")
43 | (poly2grau 1 2 1) ;passa para a função calcular usando a formula de bhaskara (a = 1, b = 2 e c = 1)
44 | (display "deveria ser -1")
45 | (newline)
46 |
47 | (display "3X^2 + 4X + 1 = 0 , Raizes = ")
48 | (poly2grau 3 4 1) ;passa para a função calcular usando a formula de bhaskara (a = 3, b = 4 e c = 1)
49 | (display "deveria ser -1/3 -1")
50 | (newline)
51 |
52 | (newline)
53 | (display "2X^2 + 4X + 3 = 0 , Raizes = ")
54 | (poly2grau 2 4 3) ;passa para a função calcular usando a formula de bhaskara (a = 2, b = 4 e c = 3)
55 | (display "deveria ser Nenhuma")
56 | (newline)
57 |
58 | (newline)
59 | (display "X^2 -1 = 0 , Raizes = ")
60 | (poly2grau 1 0 -1) ;passa para a função calcular usando a formula de bhaskara (a = 1, b = 0 e c = -1)
61 | (display "deveria ser 1 e -1")
62 | (newline)
63 |
64 | (newline)
65 | (display "2X^2 +4X +2 = 0 , Raizes = ")
66 | (poly2grau 2 4 2) ;passa para a função calcular usando a formula de bhaskara (a = 2, b = 4 e c = 2)
67 | (display "deveria ser -1")
68 | (newline)
69 |
70 | (newline)
71 | (display "X + 1 = 0 , Raizes = ")
72 | (poly2grau 0 1 1) ;passa para a função calcular usando a formula de bhaskara (a = 0, b = 1 e c = 1)
73 | (display "deveria ser: Degenerada")
74 | (newline)
75 |

```

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)

Aluno(a): Lucas Nathã Monteiro Pimentel de Souza

16.1. Fazer testes para outros cinco polinômios de segundo grau

```

39 ;;-----
40 ;Exercicio da lista 16.1
41 (display "X^2 + X + 1 = 0 , Raizes = ")
42 (poly2grau 1 1 1)
43 (newline)
44 (display "2X^2 + X + 3 = 0 , Raizes = ")
45 (poly2grau 2 1 3)
46 (newline)
47 (display "X^2 +2 = 0 , Raizes = ")
48 (poly2grau 1 0 2)
49 (newline)
50 (display "2X^2 +4X +2 = 0 , Raizes = ")
51 (poly2grau 2 4 2)
52 (newline)
53 (display "X + 5 = 0 , Raizes = ")
54 (poly2grau 0 1 5)
55
56
57
58
59
60
61 (newline)
62 (newline)

```

Welcome to [DrRacket](#), version 8.12 [cs].

Language: [Advanced Student](#) [custom]; memory limit: 128 MB.

Teachpack: [gui.rkt](#).

```

    UENF-CCT-LCMAT-CC, 2024
    Paradigmas de Linguagens de Programacao (Prof. Ausherto Castro)
    Aluno(a): Lucas Nathã Monteiro Pimentel de Souza
X^2 + X + 1 = 0 , Raizes = "Nenhuma Ou Complexa"

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

X^2 +2 = 0 , Raizes = "Nenhuma Ou Complexa"

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

X + 5 = 0 , Raizes = "degenerada"

```

Arquivo [20-estruturas.rtk](#) Aplicações: Estruturas de dados

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

```

2  ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3  ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4  ;; 2024
5  ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== seu nome aqui e abaixo
6  ;;
7  ;; define a linguagem default ==> Habilite Advanced Student
8  ; -----
9  (newline)
10 (display " UENF-CCT-LCMAT-CC, 2024")
11 (newline)
12 (display " Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
13 (newline)
14 (display " Aluno(a): Lucas Nathã Monteiro Pimentel de Souza")
15 (newline)
16 ;;
17 ; Ir no Menu:
18 ; Language -> Choose Language -> How to Design Programs -> Advanced Student
19 ; -----
20 (define-struct entrada (nome cep fone)) ;define uma estrutura com 3 variaveis (nome, cep e fone)
21
22 (newline)
23 (make-entrada 'Antoniolopes '05890-001 '606-7771) ;coloca o nome Antoniolopes na variavel nome, 05890-001 no cep e 606-7771 no fone. tudo isso na variavel make-entrada
24 (newline)
25 (display "Nome = " )
26 (entrada-nome (make-entrada 'Antoniolopes '05890-001 '606-7771)) ;imprime o nome
27 (display "CEP = " )
28 (entrada-cep (make-entrada 'Antoniolopes '05890-001 '606-7771)) ;imprime o cep
29 ;;
30 (define abc (make-entrada 'Antoniolopes '05890-001 '606-7771))
31 (display "Fone = " )
32 (entrada-fone abc) ;imprime o fone

```

```

UENF-CCT-LCMAT-CC, 2024
Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
Aluno(a): Lucas Nathã Monteiro Pimentel de Souza

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

Nome = 'Antoniolopes
CEP = '05890-001
Fone = '606-7771

```

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

```

8 | ; -----
9 | (newline)
10 | (display " UENF-CCT-LCMAT-CC, 2024")
11 | (newline)
12 | (display " Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
13 | (newline)
14 | (display " Aluno(a): Lucas Nathã Monteiro Pimentel de Souza")
15 | (newline)
16 | ;;
17 | ; Ir no Menu:
18 | ; Language -> Choose Language -> How to Design Programs -> Advanced Student
19 | ; -----
20 | (define-struct pc (processador placadevideo))
21 | (make-pc 'Ryzen 'Gtx)
22 | (display "Processador = ") (pc-processador (make-pc 'Ryzen 'Gtx))
23 | (display "Placa de video = ") (pc-placadevideo (make-pc 'Ryzen 'Gtx)) (newline)
24 |
25 | (define-struct celular (android mac))
26 | (make-celular 'xiaomi 'iphone)
27 | (display "Android = ") (celular-android (make-celular 'xiaomi 'iphone))
28 | (display "Mac = ") (celular-mac (make-celular 'xiaomi 'iphone)) (newline)
29 |
30 | (define-struct serhumano (superior inferior))
31 | (make-serhumano 'braço 'perna)
32 | (display "Superior = ") (serhumano-superior (make-serhumano 'braço 'perna))
33 | (display "Inferior = ") (serhumano-inferior (make-serhumano 'braço 'perna)) (newline)
.. |

```

```

UENF-CCT-LCMAT-CC, 2024
Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
Aluno(a): Lucas Nathã Monteiro Pimentel de Souza
(make-pc 'Ryzen 'Gtx)
Processador = 'Ryzen
Placa de vídeo = 'Gtx

(make-celular 'xiaomi 'iphone)
Android = 'xiaomi
Mac = 'iphone

(make-serhumano 'braço 'perna)
Superior = 'braço
Inferior = 'perna

```

Arquivo [30-entrada.rtk](#)

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

```

11 (newline)
12 (display "  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
13 (newline)
14 (display "  Aluno(a): Lucas Nathã Monteiro Pimentel de Souza")
15 (newline)
16 ;;
17 (newline)
18 (newline)
19
20 (display "Escreva QUATRO notas (numeros menores que 10, separados por espaços a b c d) : ")
21 (newline)
22 (define a (read)) ;vai ler oque o usuário digitar e botar um em cada variavel
23 (define b (read))
24 (define c (read))
25 (define d (read))
26
27 (define media
28   (lambda (x y z w)
29     (/ (+ (+ (+ x y) z) w) 4) ;calculo da média aritmetica
30   ))
31
32 (display "A media = ")
33 (media a b c d)
34
35 (if (> (media a b c d) 5) ;se media > 5 então aprovado, senão reprovado
36     (display "Aprovado")
37     (display "Reprovado"))
--

```

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)

Aluno(a): Lucas Nathã Monteiro Pimentel de Souza

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

5

8

9

3

A media = $6\frac{1}{4}$

Aprovado

- 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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== seu nome aqui e abaixo
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (newline)
10 (display "  UENF-CCT-LCMAT-CC, 2024")
11 (newline)
12 (display "  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
13 (newline)
14 (display "  Aluno(a): Lucas Nathã Monteiro Pimentel de Souza")
15 (newline)
16 ;;
17 (display "Escreva o nome e a idade (separados por espaços) : ")
18 (newline)
19 (define nome1 (read))
20 (define idade1 (read))
21 (display "Escreva o nome e a idade da segunda pessoa (separados por espaços) : ")
22 (define nome2 (read))
23 (define idade2 (read))
24
25 (display "Nome da primeira pessoa = ") nome1
26 (display "Idade da primeira pessoa = ") idade1
27 (newline)
28 (display "Nome da segunda pessoa = ") nome2
29 (display "Idade da segunda pessoa = ") idade2
30

```

```

  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno(a): Lucas Nathã Monteiro Pimentel de Souza
Escreva o nome e a idade (separados por espaços) :
Lucas
24
Escreva o nome e a idade da segunda pessoa (separados por espaços) : Yan
22
Nome da primeira pessoa = 'Lucas
Idade da primeira pessoa = 24

Nome da segunda pessoa = 'Yan
Idade da segunda pessoa = 22

```

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 Racket (Scheme)
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6 ;;
7 ;;;;;;;;;;;;;; Escolha a linguagem "Determine language from source"
8 ;;
9
10 ; -----
11 (display "    UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display "    Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display "    Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 (newline)
17 ;;
18
19 (define (conta a b)
20   (let ((n1 (/ (+ 3 a) (- (* 7 b) 2)))
21         (n2 (- (* a b) 9)))
22     (- n1 n2))
23 )
24
25 (display (conta 5 1))
26
27
28
29
```

```
Welcome to DrRacket, version 8.12 [cs].
Language: Advanced Student; memory limit: 128 MB.
    UENF-CCT-LCMAT-CC, 2024
    Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
    Aluno: Lucas Nathã Monteiro Pimentel de Souza
28/5
>
```

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


```

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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6 ;;
7 ;;:::::::::::::::::::: Escolha a linguagem "Determine language from source"
8 ;;
9
10 ; -----
11 (display "      UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display "      Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display "      Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 (newline)
17 ;;
18
19 (define (raiz a)
20   (sqrt (- (+ (* a a) (+ 3 a)) 5))
21 )
22
23 (display (raiz 2))
24
25
26
27
28
29

```

Welcome to [DrRacket](#), version 8.12 [cs].

Language: **Advanced Student**; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2024

Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)

Aluno: Lucas Nathã Monteiro Pimentel de Souza

2

>

- Criar uma lista com 5 elementos

```

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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6 ;;
7 ;;:::::::::::::::::::: Escolha a linguagem "Determine language from source"
8 ;;
9
10 ; -----
11 (display "      UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display "      Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display "      Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 (newline)
17 ;;
18
19 (define lista '(1 2 3 4 5))
20
21 (display "Lista = ") lista
22
23
24
25
26
27
28
29

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: Advanced Student; memory limit: 128 MB.
      UENF-CCT-LCMAT-CC, 2024
      Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
      Aluno: Lucas Nathã Monteiro Pimentel de Souza
Lista = (list 1 2 3 4 5)
> |

```

- **Determinar o segundo elemento de uma lista**
- **Determinar o antepenúltimo elemento de uma lista dada**
- **Consultar se um elemento pertence a uma lista dada**
- **Adicionar o terceiro elemento de uma lista**
 - No final de outra lista A
 - No início de outra lista B

```

41
42 (define lista '(1 2 3 4 5))
43 (define lista2 '(10 9 8 7 6))
44
45 (display "Segundo elemento lista = ")
46 (segundo-elemento lista)
47
48 (display "Antepenultimo elemento lista = ")
49 (antepenultimo-elemento lista)
50
51 (display "Tem o elemento 2 na lista? ")
52 (elemento-na-lista? 2 lista)
53
54 (display "Adicionando o terceiro elemento: ")
55 (adicionar-terceiro-elemento lista lista2)
56
57
58
59
60
61
62
63
64
65
66
67
68
69

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: Advanced Student; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno: Lucas Nathã Monteiro Pimentel de Souza
Segundo elemento lista = 2
Antepenultimo elemento lista = 4
Tem o elemento 2 na lista? #true
Adicionando o terceiro elemento: (list 10 9 8 7 6 3)
>

```

- Calcular o perímetro de um quadrado, círculo ou triângulo

```

18
19 (define (calcular-perimetro tipo parametros)
20   (cond
21     [(equal? tipo 'quadrado)
22      (* 4 (car parametros))] ; Perímetro do quadrado: 4 * lado
23
24     [(equal? tipo 'circulo)
25      (* 2 (* pi (car parametros)))] ; Perímetro do círculo: 2 * π * raio
26
27     [(equal? tipo 'triangulo)
28      (apply + parametros)] ; Perímetro do triângulo: soma dos lados
29
30     [else
31      ((display "N existe"))]))
32
33
34
35
36 (display "Perímetro do quadrado com lado 5: ")
37 (calcular-perimetro 'quadrado '(5)) ; Saída: 20
38
39 (display "Perímetro do círculo com raio 3:")
40 (calcular-perimetro 'circulo '(3)) ; Saída: ~18.8495
41
42 (display "Perímetro do triângulo com lados 4, 5 e 6:")
43 (calcular-perimetro 'triangulo '(4 5 6)) ; Saída: 15
44
45
46
47

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: Advanced Student, memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2024
  Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
  Aluno: Lucas Nathã Monteiro Pimentel de Souza
Perímetro do quadrado com lado 5: 20
Perímetro do círculo com raio 3:~18.84955592153876
Perímetro do triângulo com lados 4, 5 e 6:15
>

```

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

```

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 ;; 2024
5 ;; Aluno: Lucas Nathã Monteiro Pimentel de Souza <===== escreva seu nome aqui
6 ;;
7 ;;:::::::::::::::::::: Escolha a linguagem "Determine language from source"
8 ;;
9
10 ; -----
11 (display "      UENF-CCT-LCMAT-CC, 2024")
12 (newline)
13 (display "      Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)")
14 (newline)
15 (display "      Aluno: Lucas Nathã Monteiro Pimentel de Souza")
16 (newline)
17 ;;
18
19 (define (kesimo x)
20   (cond
21     ((even? x) (* x (/ x 2)))
22     ((odd? x) (+ 1 (* x (/ (- x 1) 2))))))
23
24 (display "0 2-ésimo é = ")
25 (kesimo 2)
26 (display "0 7-ésimo é = ")
27 (kesimo 10)
28
29
30

```

```

Welcome to DrRacket, version 8.12 [cs].
Language: Advanced Student; memory limit: 128 MB.
      UENF-CCT-LCMAT-CC, 2024
      Paradigmas de Linguagens de Programacao (Prof. Ausberto Castro)
      Aluno: Lucas Nathã Monteiro Pimentel de Souza
0 2-ésimo é = 2
0 7-ésimo é = 50
>

```