|  | **Disciplina**: **AARE Paradigmas de Linguagens de Programação**  **Professor**: Prof. Ausberto S. Castro V.  **E-mail:** ascv@uenf.br  **Data**: 10 de abril de 2023 |
| --- | --- |



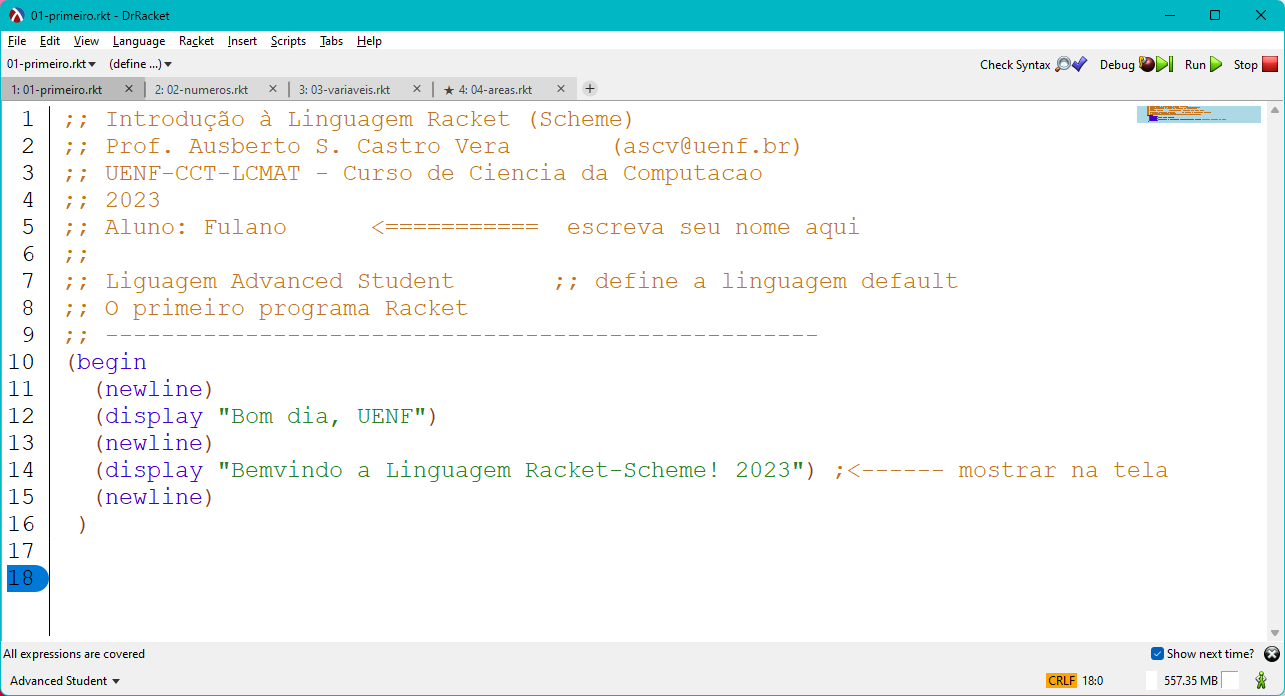
**Prática – Racket**

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?

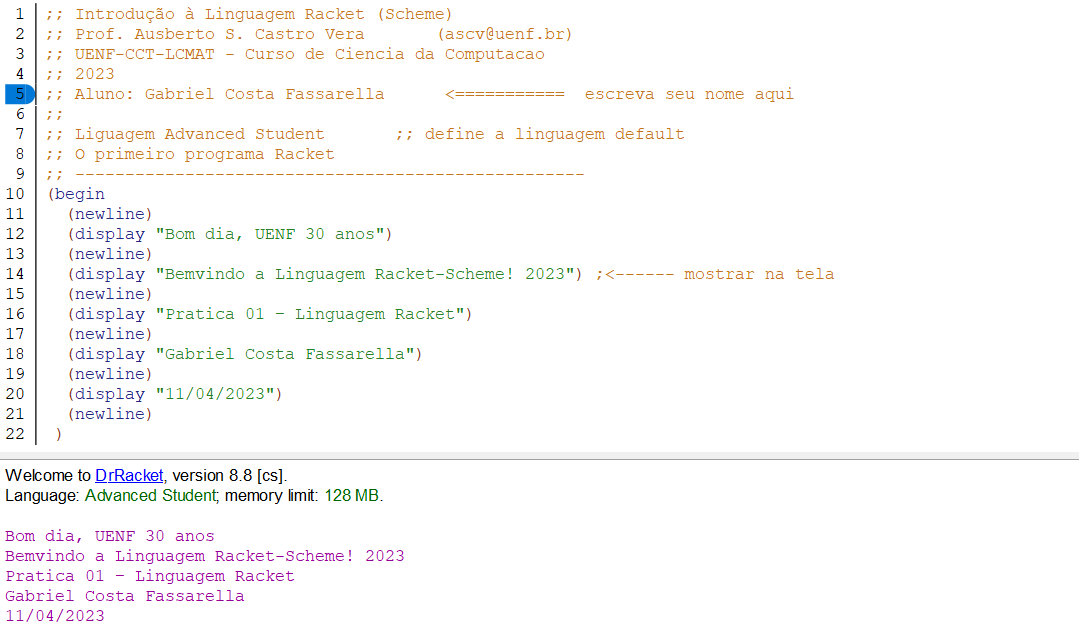


Codigo fonte

Interações (shell)

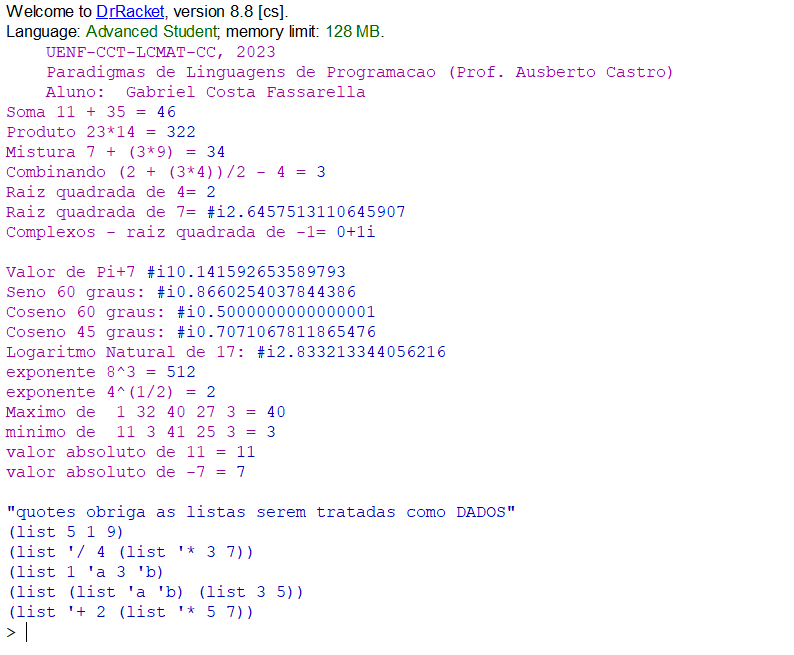
Explicar aqui: A linha 10 apresenta o comando “(begin” que indica o início do código, as linhas 11, 13 e 15 apresentam o comando “(newline)”, que indica uma quebra de linhas, e as linhas 12 e 14 apresentam o comando “(display)” seguido de uma string no interior dos parênteses, indicando que essa string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

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

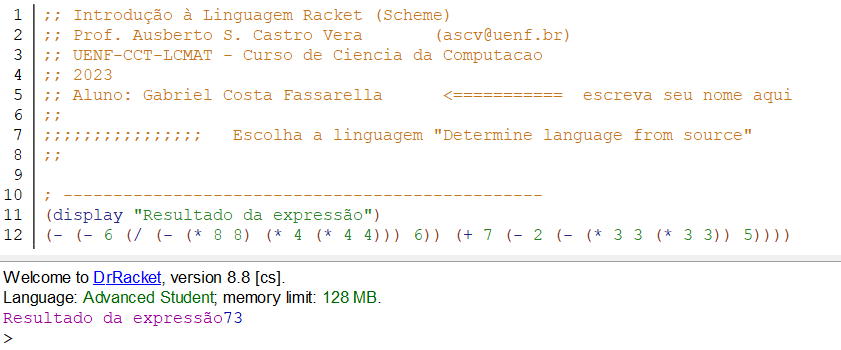


Arquivo 02-numeros.rtk Números e Aritmética

1. Execute o programa e mostre os resultados

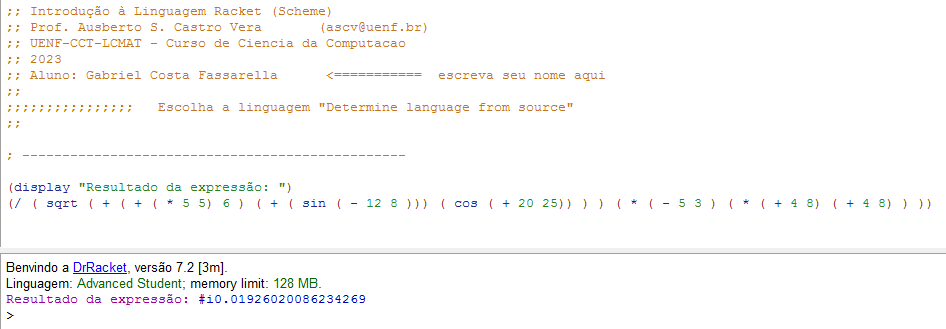


1. Escreva programas Racket para as seguintes expressões:
   1. P = (6 – (82 - 43) / 6) – (7 + (2 – (34 – 5)



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



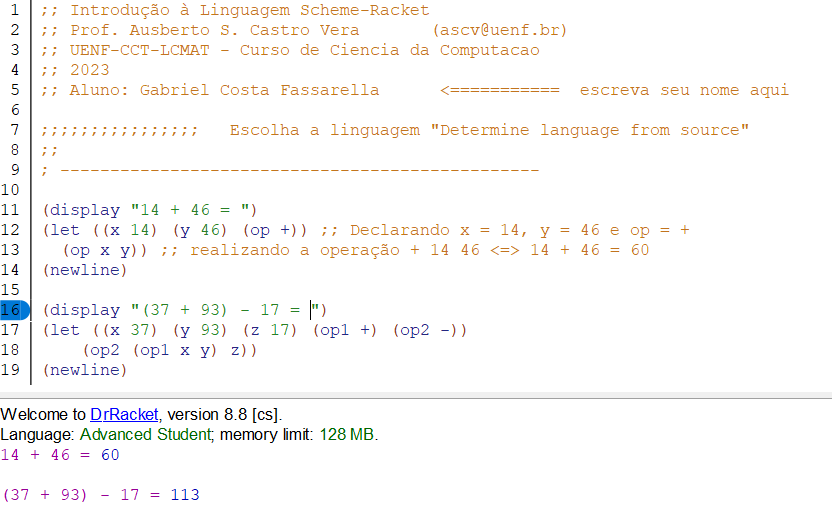


Arquivo 03-variaveis.rtk

1. Execute o programa e indique o valor das variáveis m, z, k
   1. Explicar o significado de cada uma das 3 expressões de iteração let

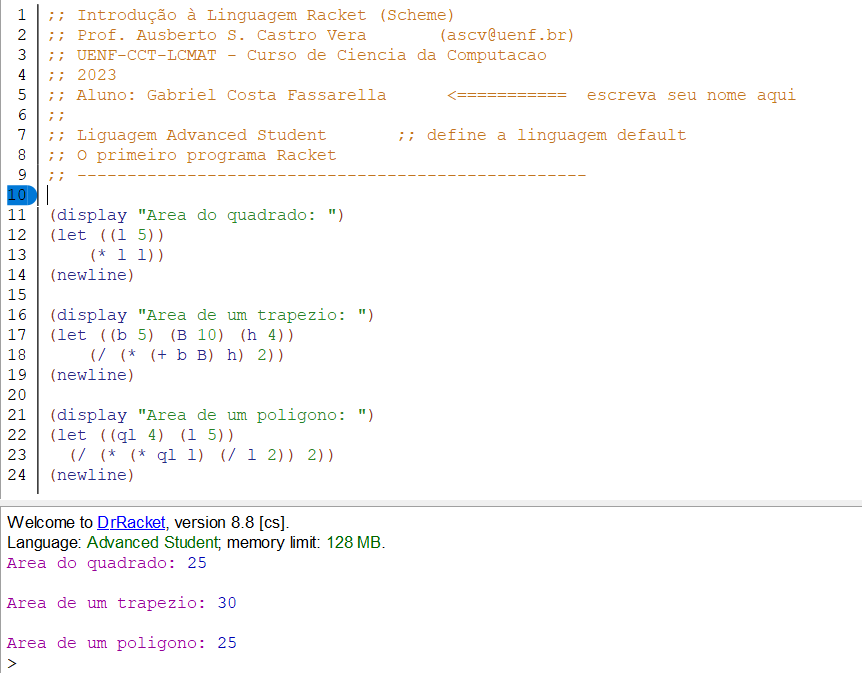
A expressão (let ((x 24)) altera o valor de x para 24 temporariamente, somando assim o valor de x com 6. O mesmo ocorre na segunda expressão, alterando o valor de a e b. Já a terceira ele muda o valor de op1 e op2 para uma operação, e o de x para um valor.

* 1. Escreva 2 expressões do tipo let e explique o seu significado

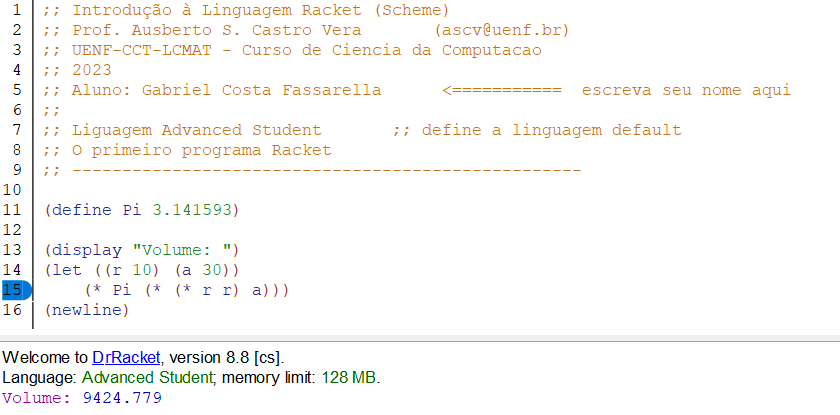


Arquivo 04-areas.rtk

1. Execute o programa e indique o que faz o programa
   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. Escreva um programa NOVO completo para calcular o volume de um galão de óleo utilizando a fórmula V = πR2A, onde as variáveis V, R e A representam, respectivamente, o volume, o raio e a altura



Arquivo 05-funcoes.rtk

1. 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.

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

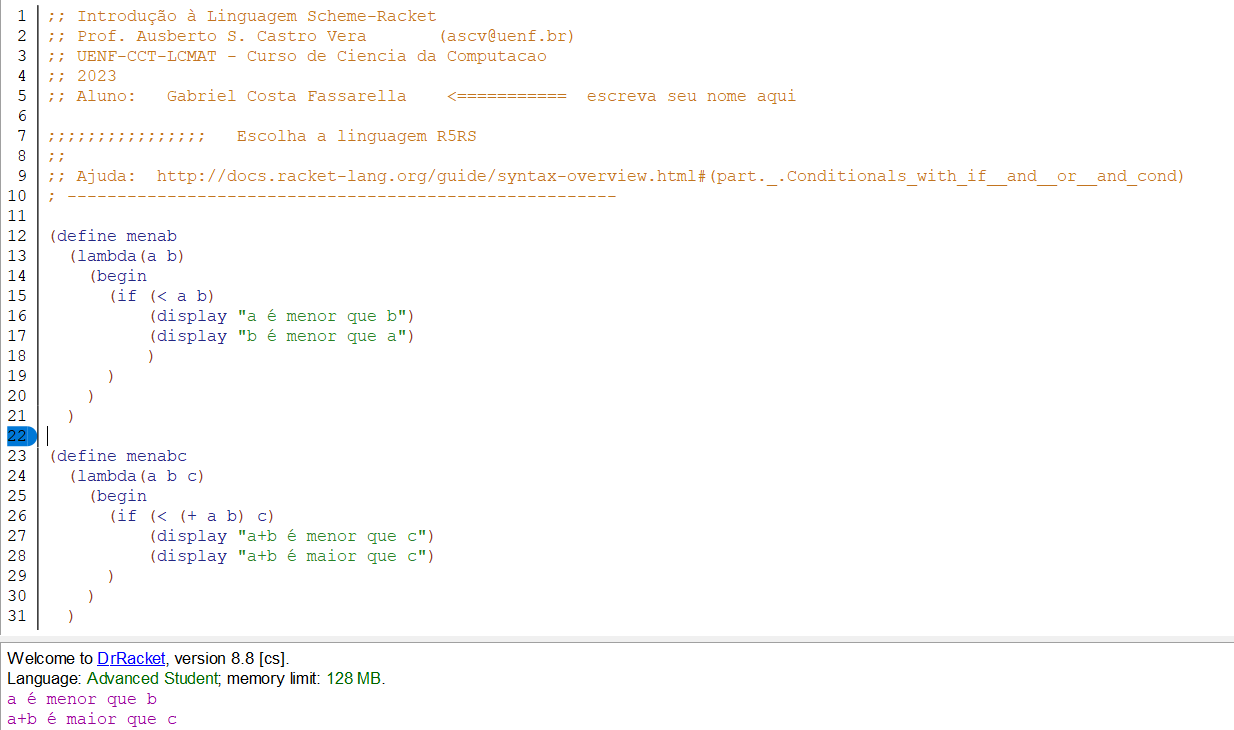


Arquivo 06-condicionalIF.rtk

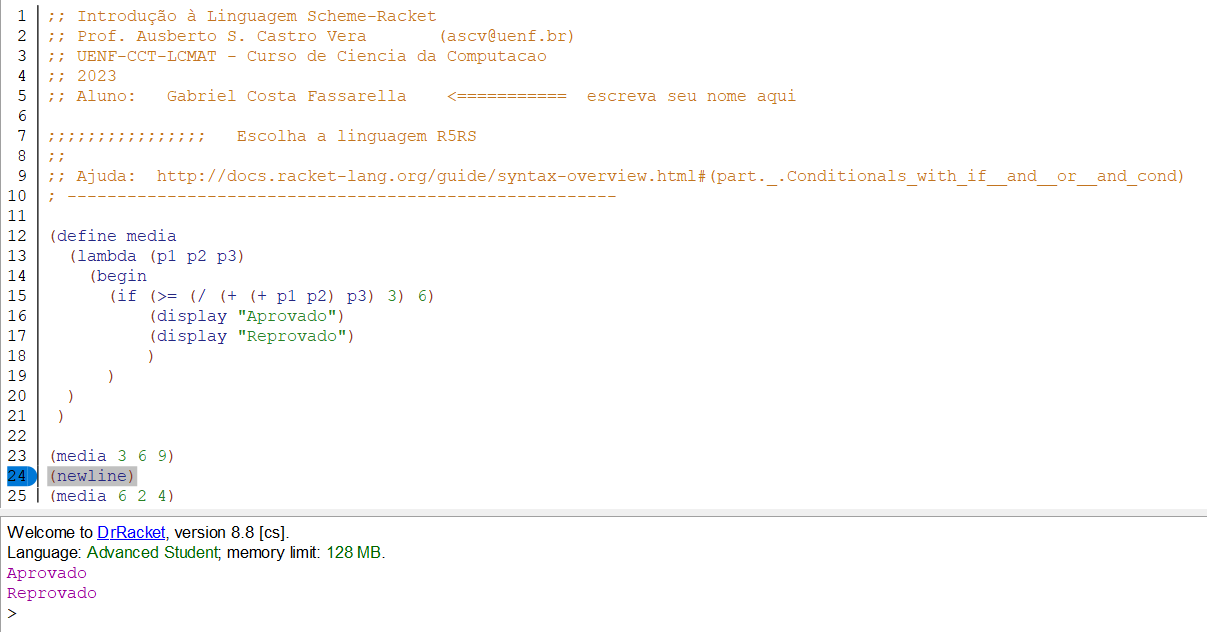
1. Executar e explicar o programa

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.

* 1. Escreva um programa com dois condicionais



* 1. 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

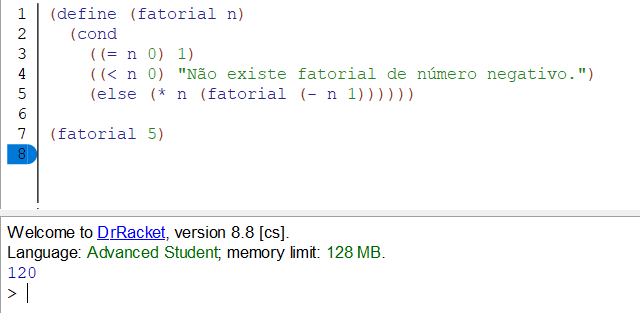


Arquivo 07-formulas.rtk

1. 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.

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

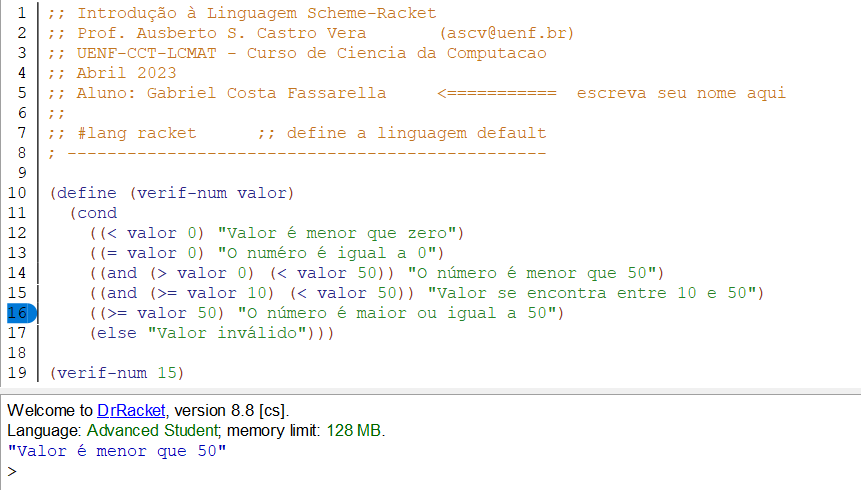


Arquivo 08-condicional.rtk

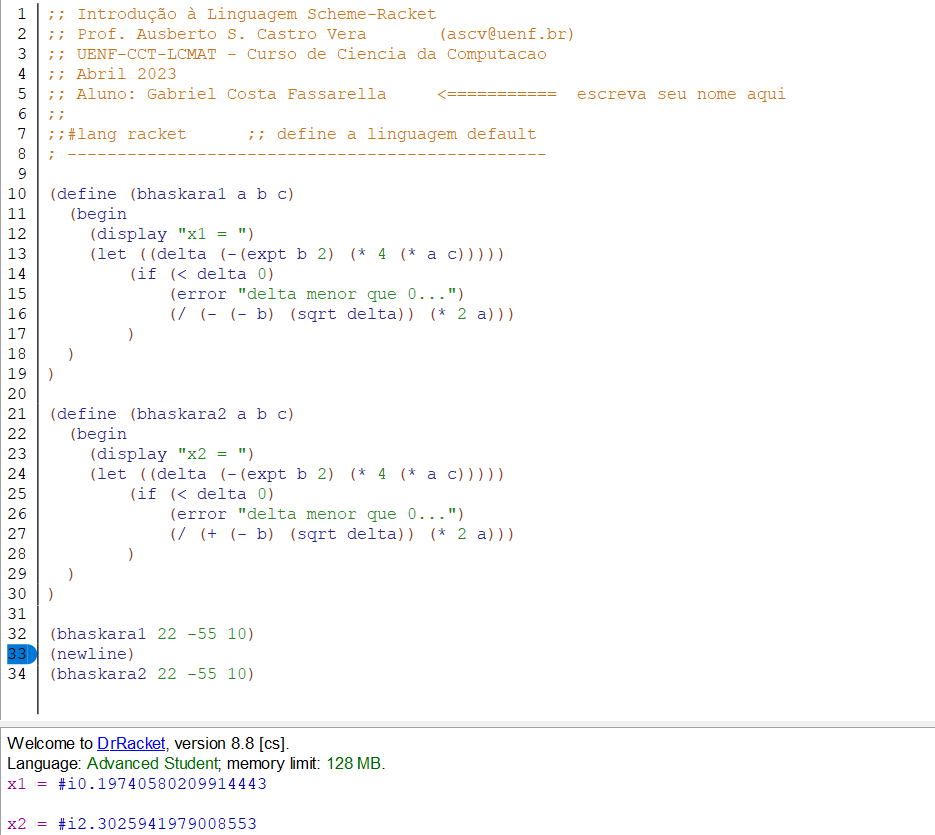
1. Execute o programa e indique o que faz

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.

* 1. Escreva um programa condicional com pelo menos 5 opções

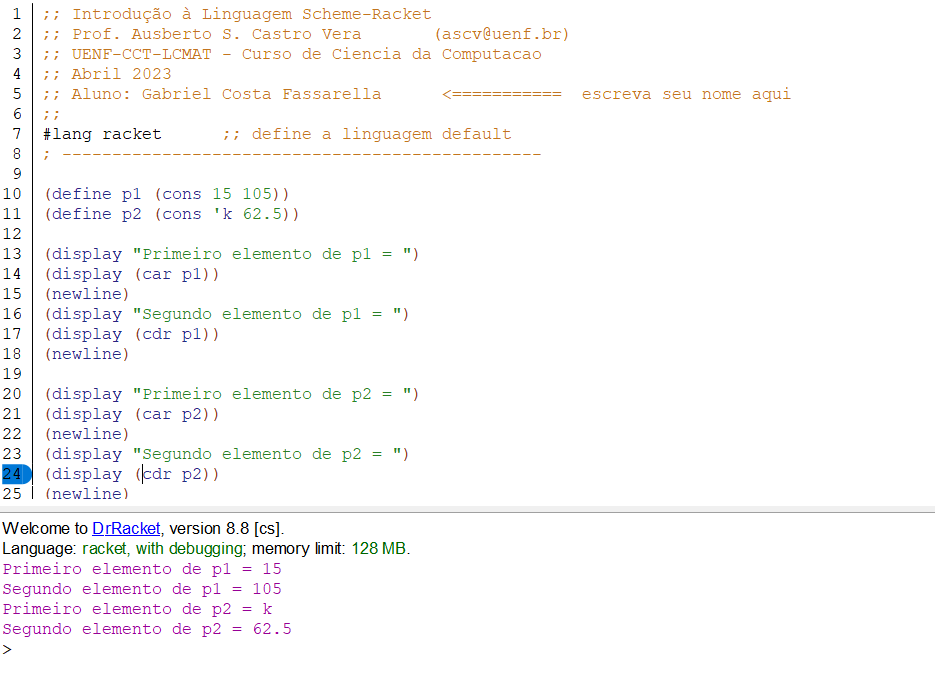


* 1. Escreva um programa bhaskara.rkt que calcule as raízes de uma equação 25x2 - 55x+10 = 0, utilizando a fórmula de Bhaskara. Sugestão: Primeiro faça o algoritmo completo



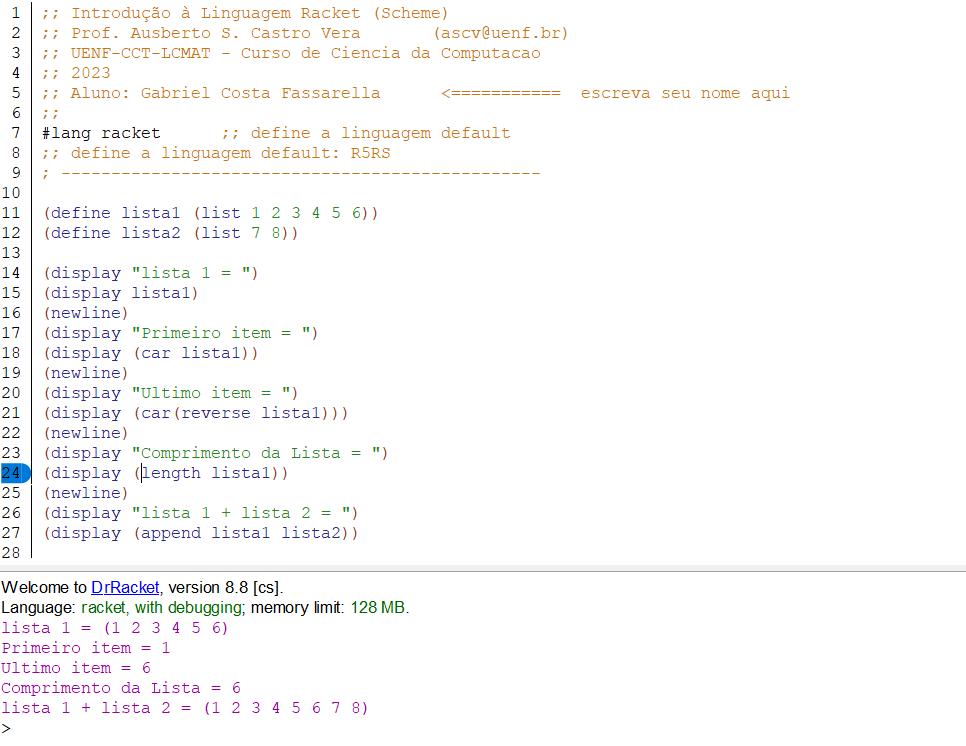
Arquivo 09-pares.rtk

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



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

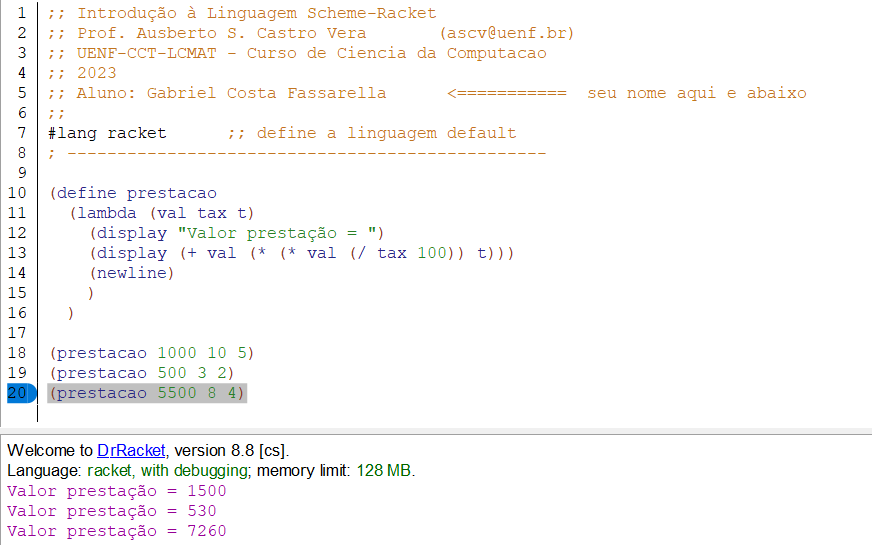
1. Executar os programas e observe quantos métodos existem para construir listas
   1. Escreva um NOVO programa para construir uma lista e determinar seu primeiro e último elemento, seu comprimento, e uma nova lista com dois elementos a mais que a anterior. Incluir os códigos fonte



* 1. 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

1. Execute os programas e indique o que faz cada um deles
   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. O que faz o seguinte procedimento **abcd**:

(define abcd

(lambda (n)

(let f ((i 2))

(cond

((>= i n) '())

((integer? (/ n i))

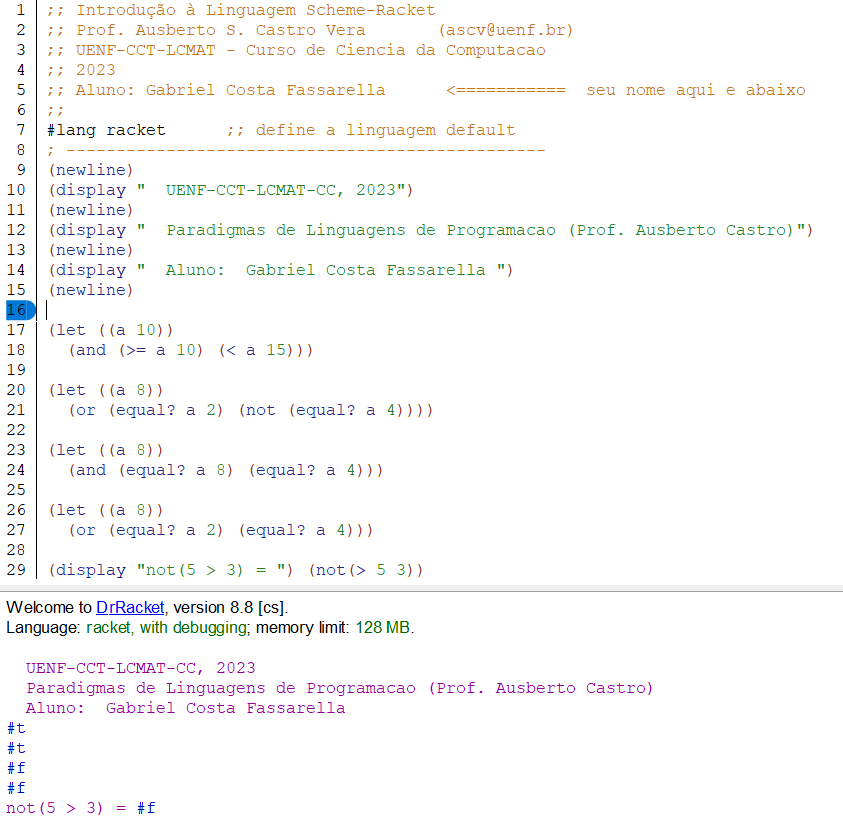
(cons i (f (+ i 1))))

(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.

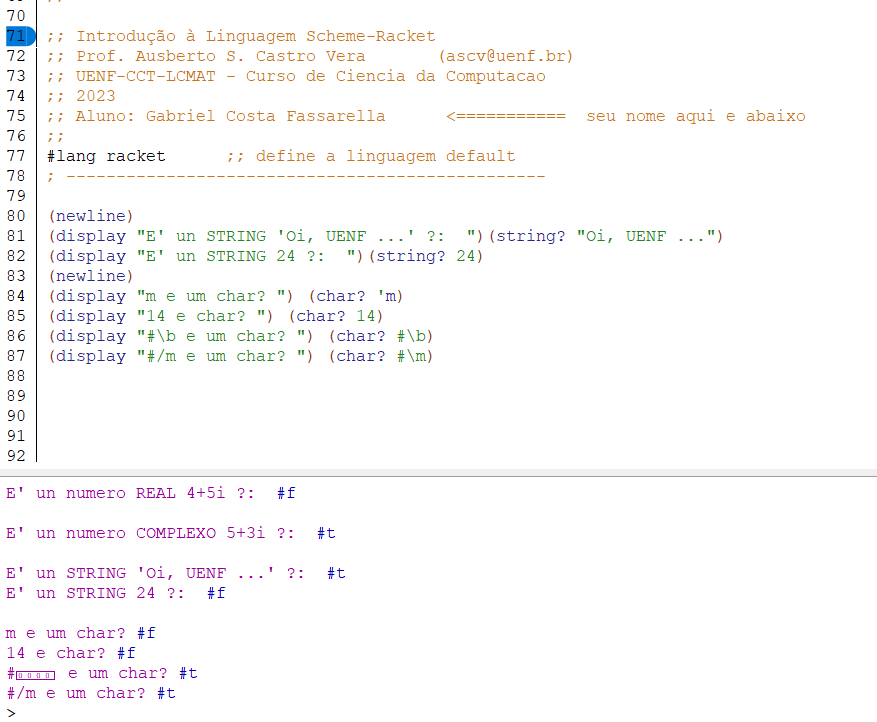
Arquivo 14-operad-logicos.rtk Operadores lógicos

1. Executar o programa e indicar o seu conteúdo
   1. Escreva e teste pelo menos cinco operações lógicas



Arquivo 15-predicados.rtk Predicados

1. Executar o programa e indicar o seu conteúdo
   1. Testar os predicados: (char? ´m) , (char? 14), (char? #\b) , (char? #\m)



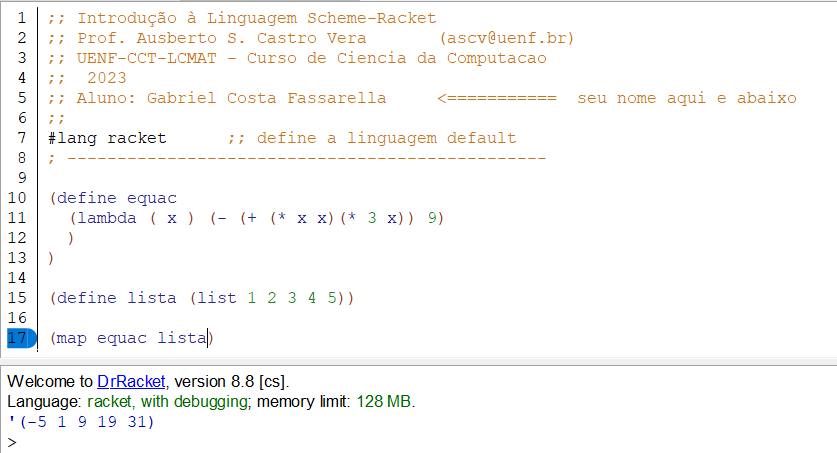
Arquivo 16-mapeamentos.rtk Mapeamentos

* 1. 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.

* 1. Construir um NOVO programa que faça o seguinte mapeamento

x ⎯→ x2 + 3x - 9

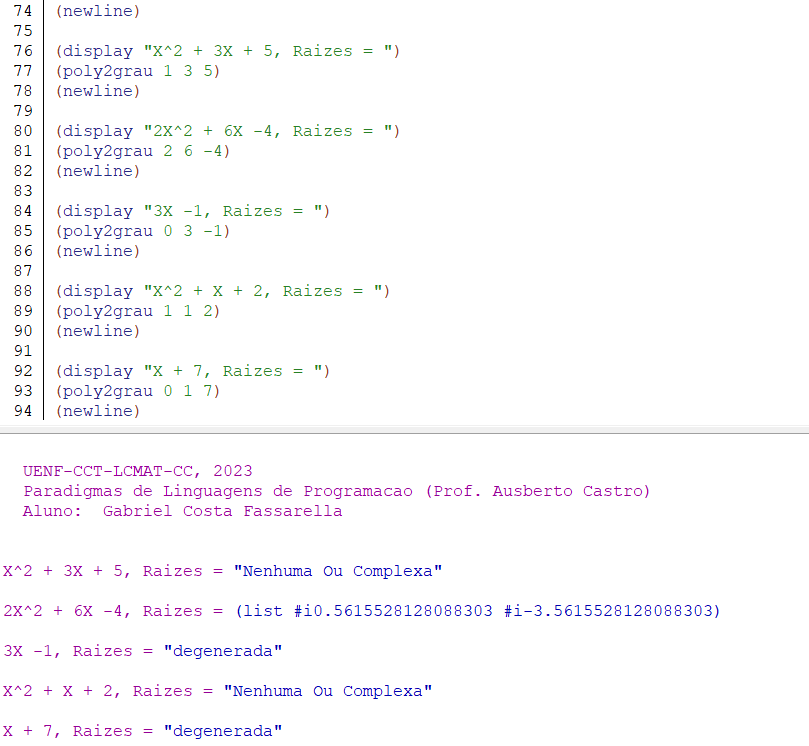


Arquivo 17-raizes-poly.rtk Aplicações: Raízes de polinômios

1. 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.

* 1. Fazer testes para outros cinco polinômios de segundo grau

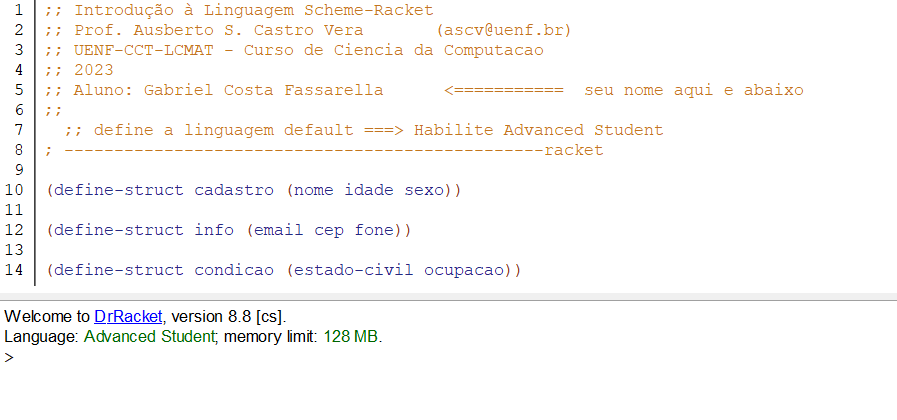


Arquivo 20-estruturas.rtk Aplicações: Estruturas de dados

1. 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.

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

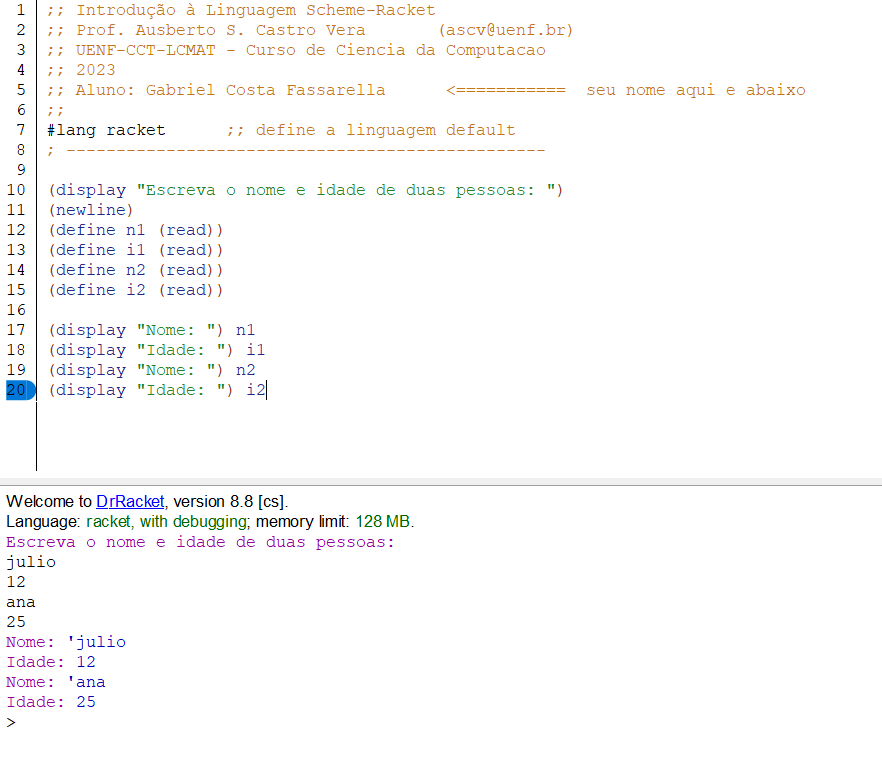


Arquivo 30-entrada.rtk

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

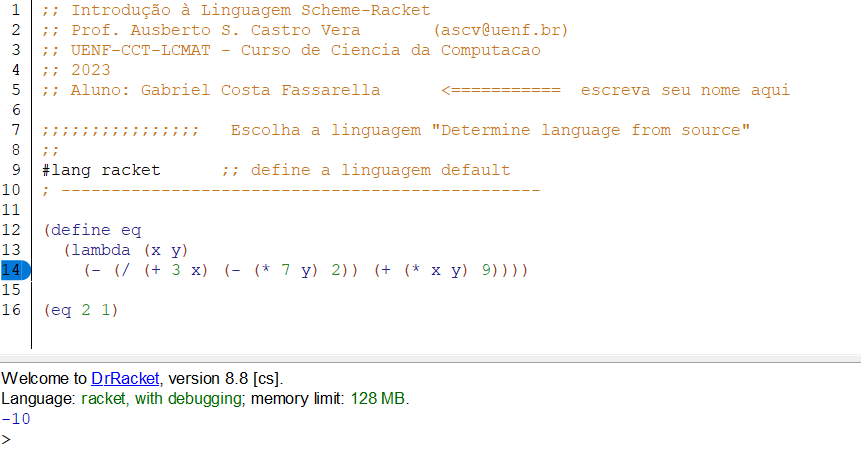
O código em questão inicia exigindo a entrada de 4 valores de entrada para o usuário, e logo em seguida realiza o cálculo de média e de acordo com o valor do cálculo devolve a situação do usuário (aprovado ou não).

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

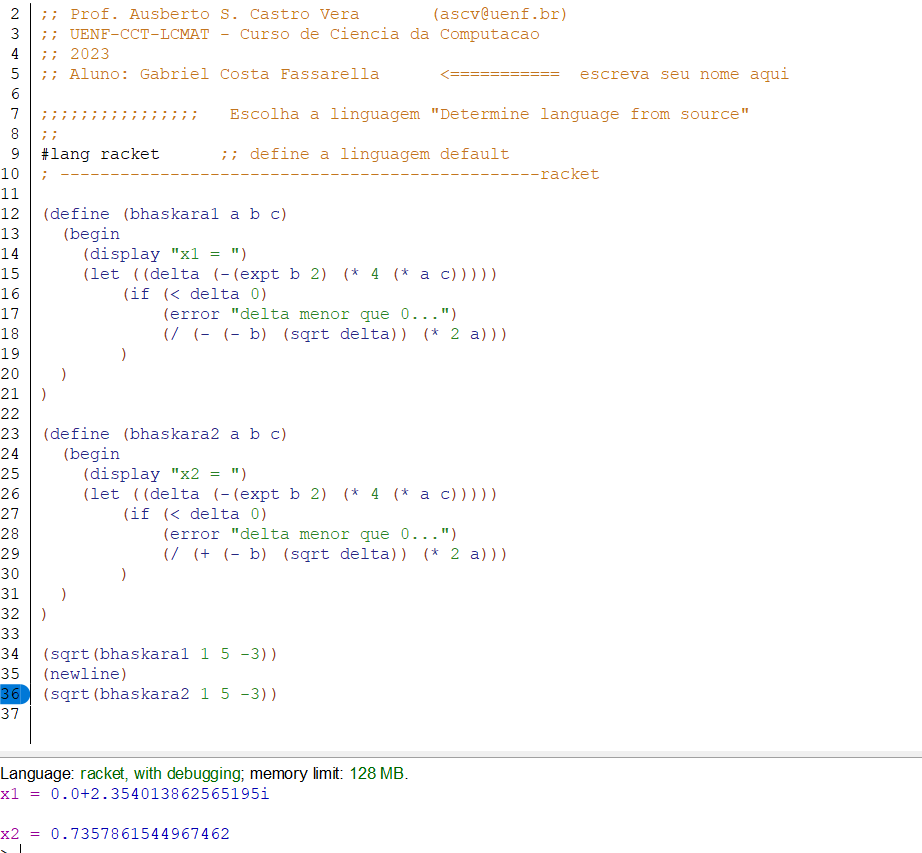


**Parte 2:**  
  
Resolver a lista de Exercícios no final da Notas de Aula (Slide 51)

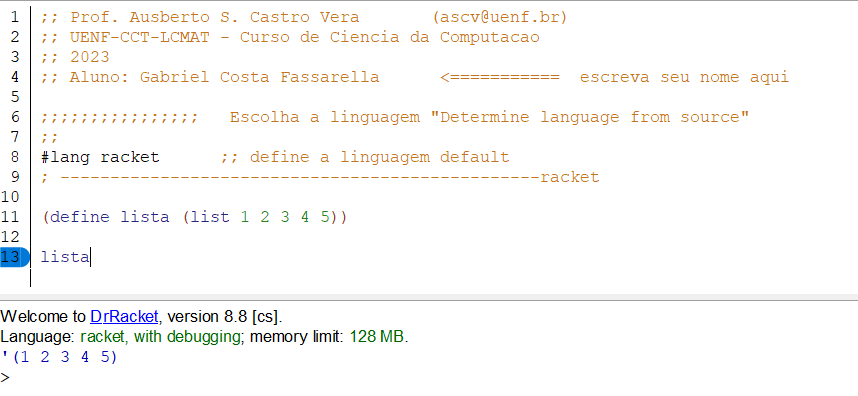
* **Programar em Racket:**
  + **(3 + x)/(7y- 2) – (xy + 9)**

****

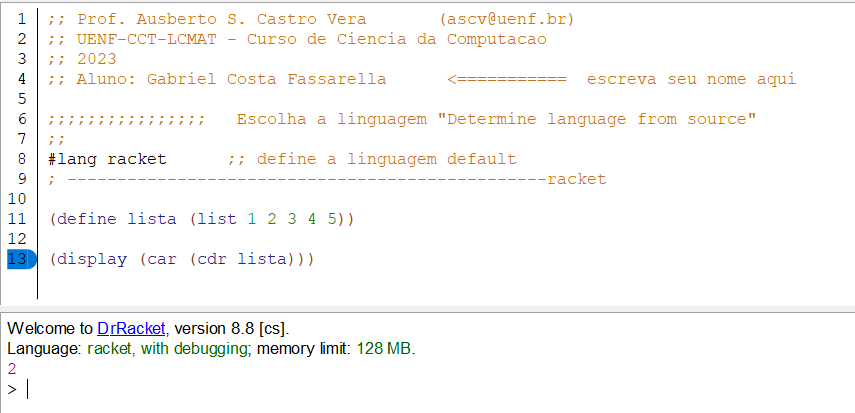
* + **Raiz quadrada de x2 + 3x -5**

****

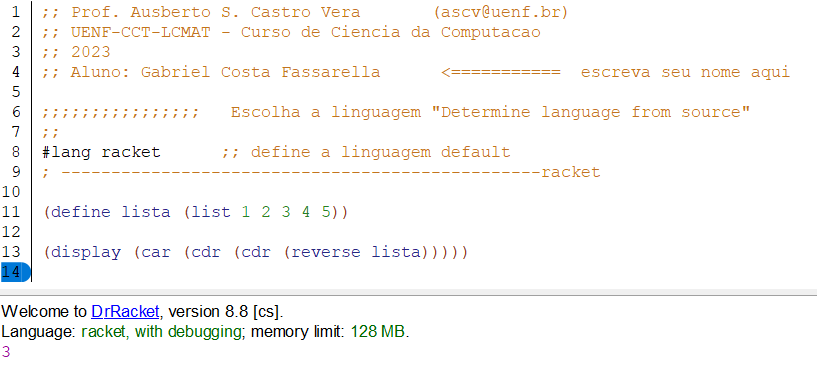
* + **Criar uma lista com 5 elementos**

****

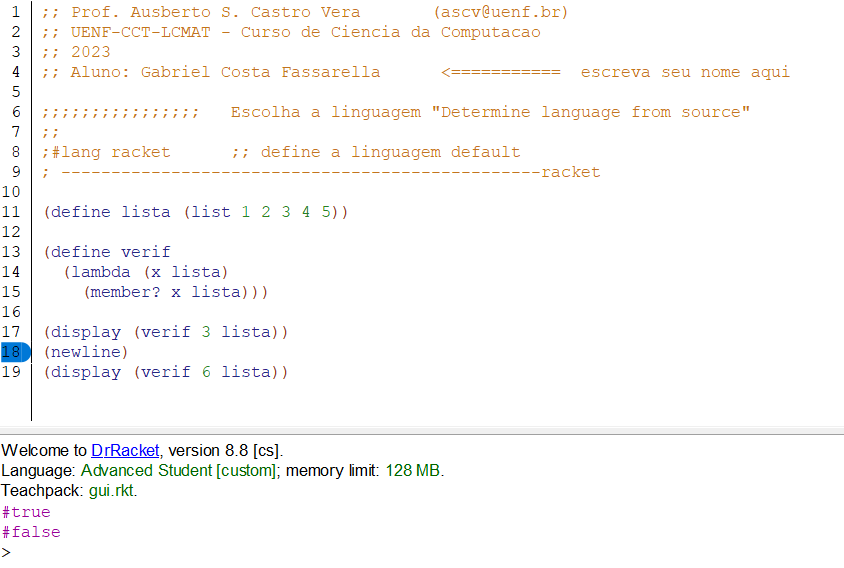
* + **Determinar o segundo elemento de uma lista**

****

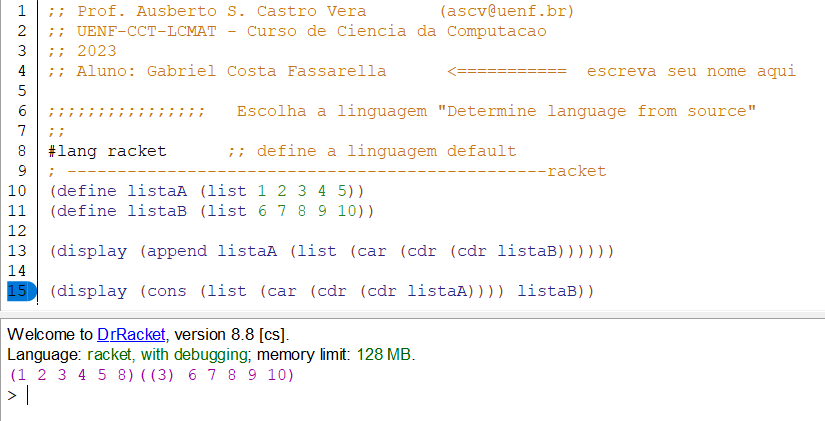
* + **Determinar o antepenúltimo elemento de uma lista dada**

****

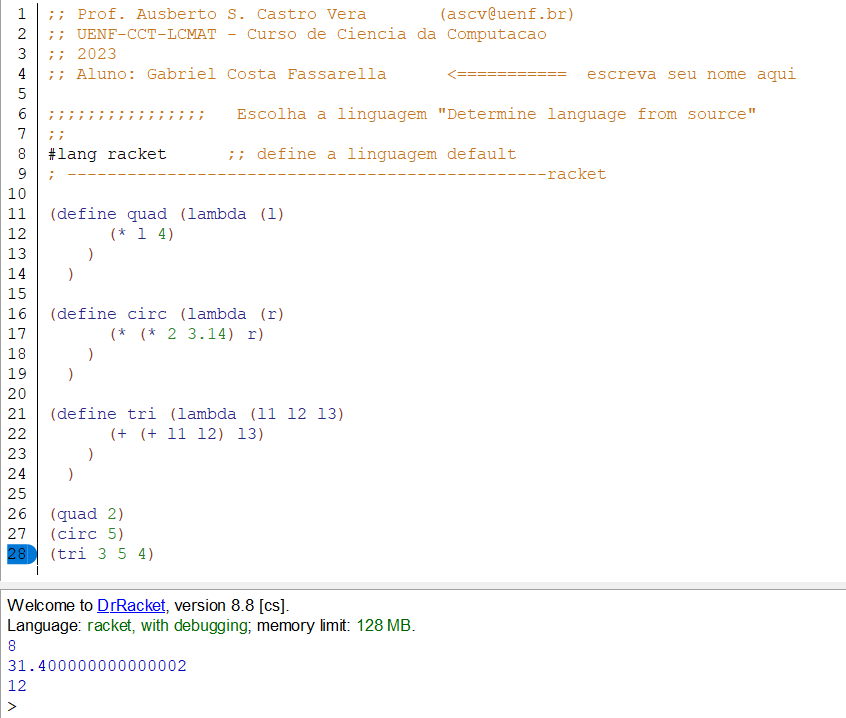
* + **Consultar se um elemento pertence a uma lista dada**

****

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

****

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

****

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

