|  | **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: Frederico Rangel Sader

Data: 10 de abril de 2023 Total Exercícios Resolvidos: .....

-------------------------✂ cortar aqui depois de ler ✂---------------------------------------------

**Instruções:**

* O **relatório** desta prática, deverá conter claramente o **Nome Completo** do aluno e a última **data** do trabalho no laboratório
* Cada item resolvido deverá conter um gráfico da **imagem capturada** da janela de execução ou janela gráfica ou do editor, onde deve aparecer obrigatoriamente o nome do aluno, e as mudanças realizadas.
* **Em todas as questões desta prática**, apresentar o código fonte modificado ou atualizado e a execução do programa (imagem capturada). Crie um **NOVO** arquivo, para cada mudança de código e mantenha o arquivo original. ArquivoN modificado deve ser renomeado para ArquivoNa, ArquivoNb, ArquivoNc, ...
* Em todos os arquivos use a linguagem R5RS, preferencialmente, exceto quando implicitamente indicado outra linguagem
* Incluir, neste relatório, os códigos fonte modificados

**Software: DrRacket** (Instale o software em idioma Inglês)

* <http://racket-lang.org/>
* <http://schemers.org/> (referências)

**Livro On-Line :** <http://www.htdp.org/> (The Book)

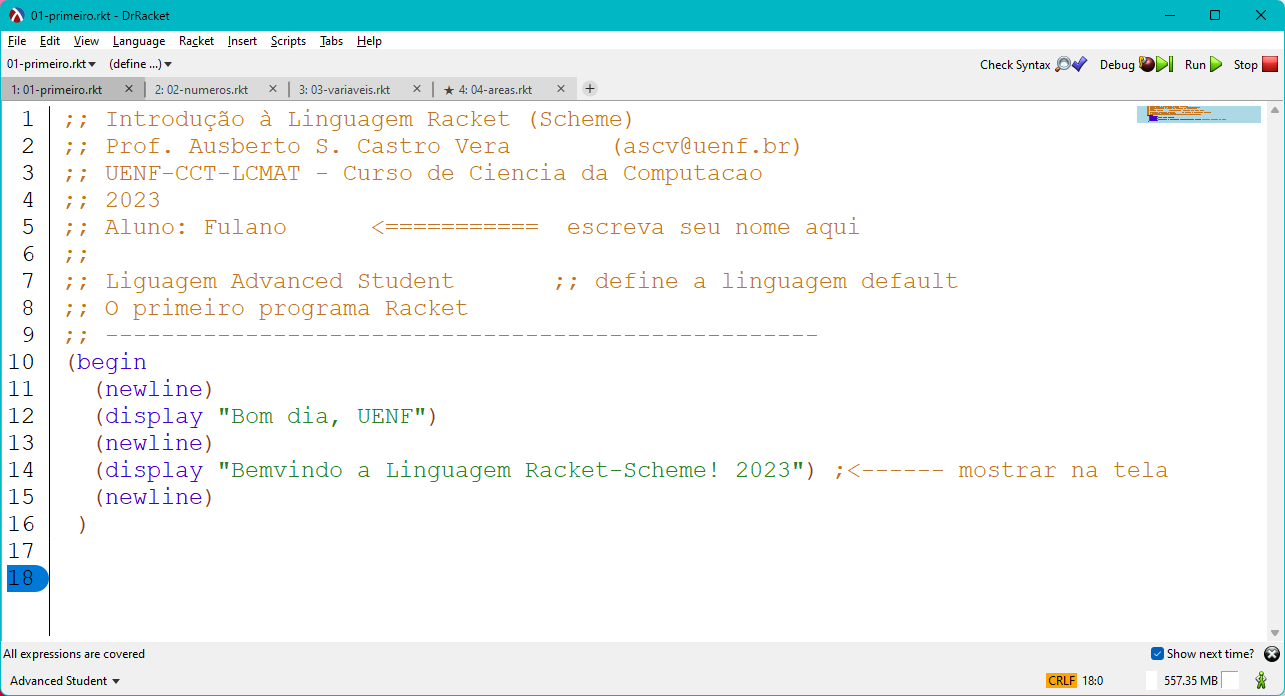
**Captura de gráficos**

* <Alt>+<Print Screen> ou
* Gadwin PrintScreen (<http://www.gadwin.com/printscreen/>)

-------------------------✂ cortar aqui depois de ler ✂------------------------------------

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:

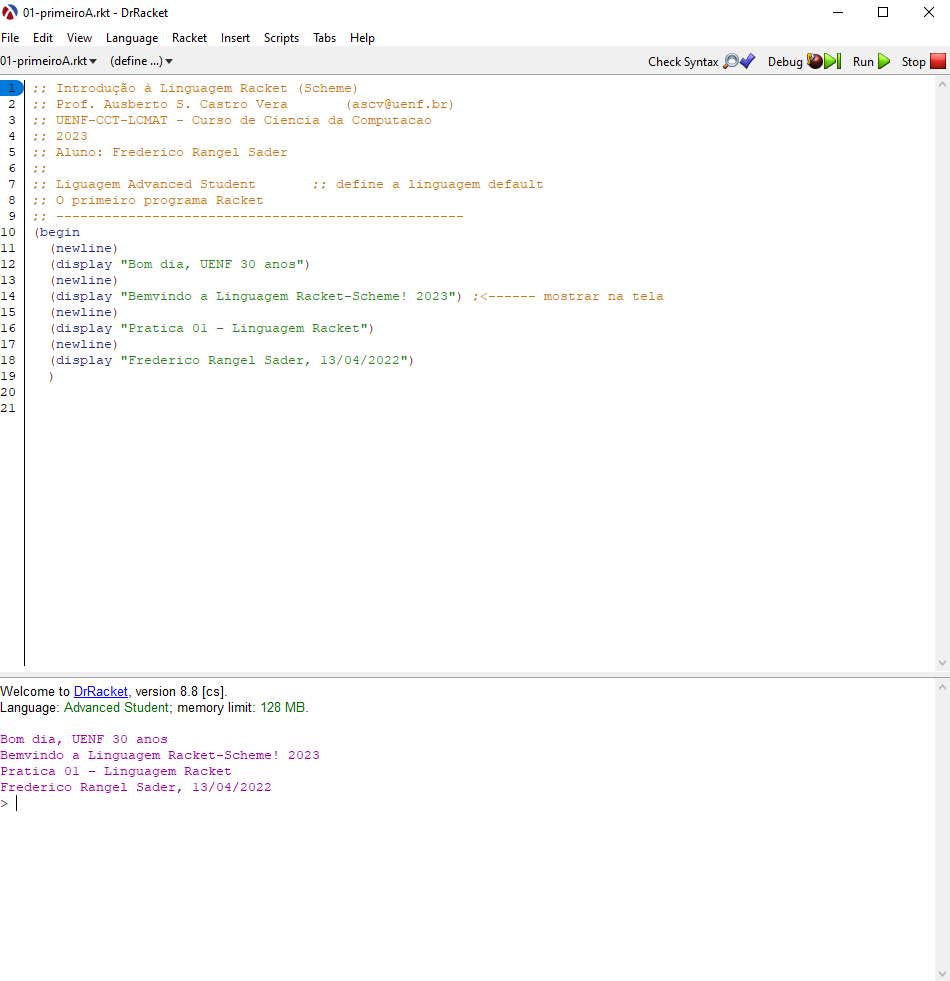
O programa imprime a seguinte mensagem na tela:

”Bom dia, UENF 30 anos

Bemvindo a Linguagem Racket-Scheme! 2023

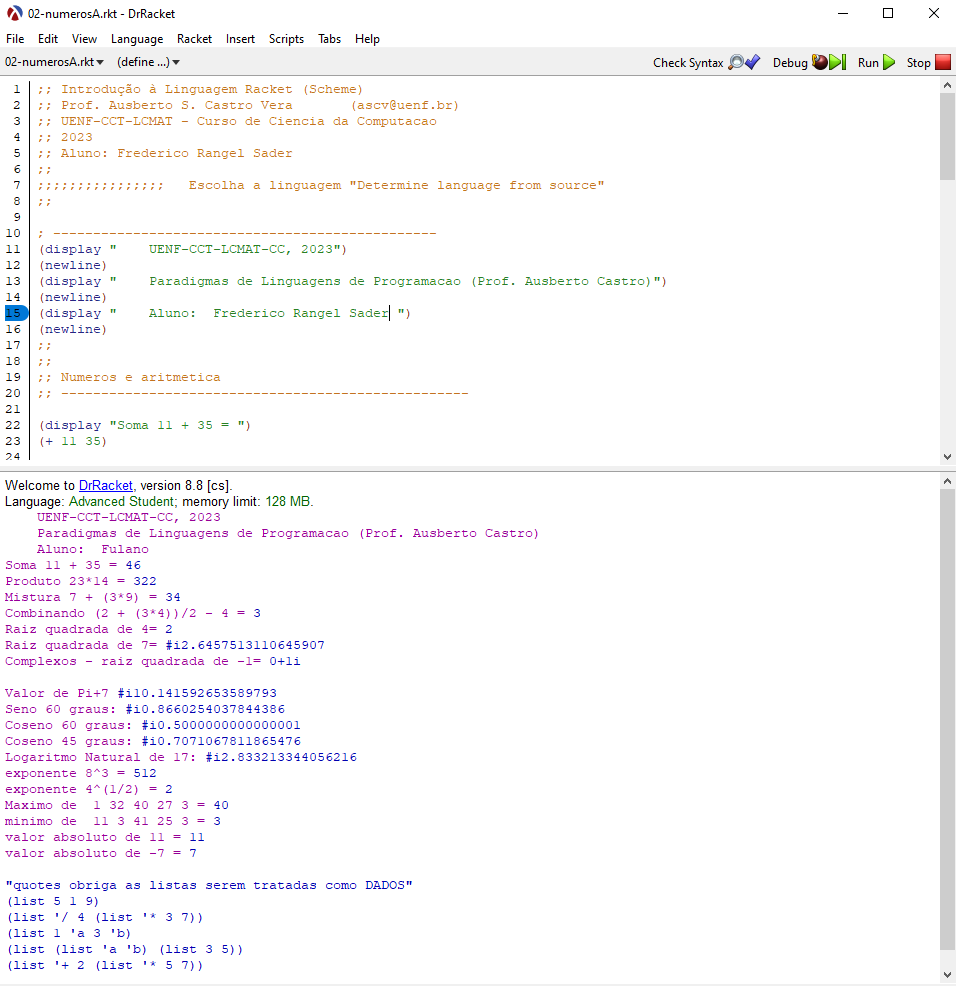
A linha 10 inicia o programa com o parêntese que é fechado na linha 16. Nas linhas 11, 13 e 15, são comandos para pular uma linha de texto. Nas linhas 12 e 14 estão os comandos para imprimir as mensagens na tela.

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

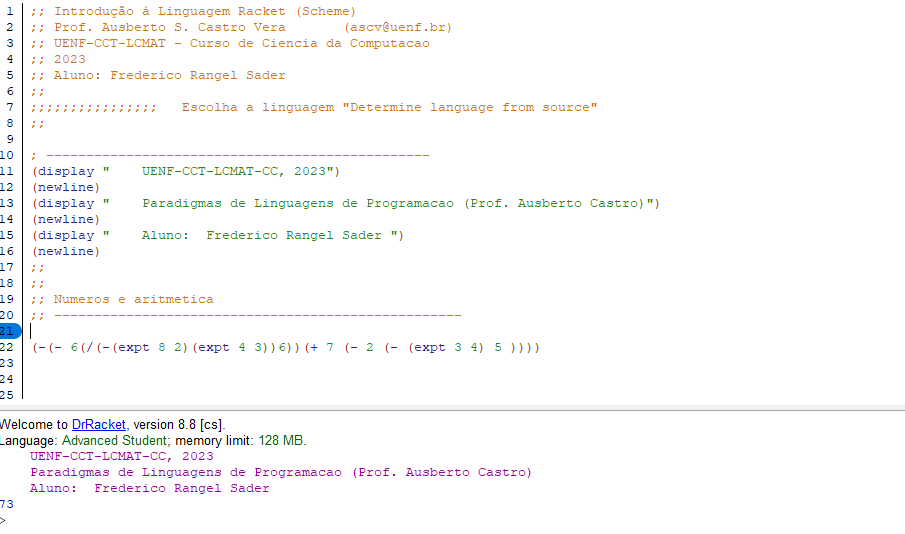


Arquivo 02-numeros.rtk Números e Aritmética

1. Execute o programa e mostre os resultados

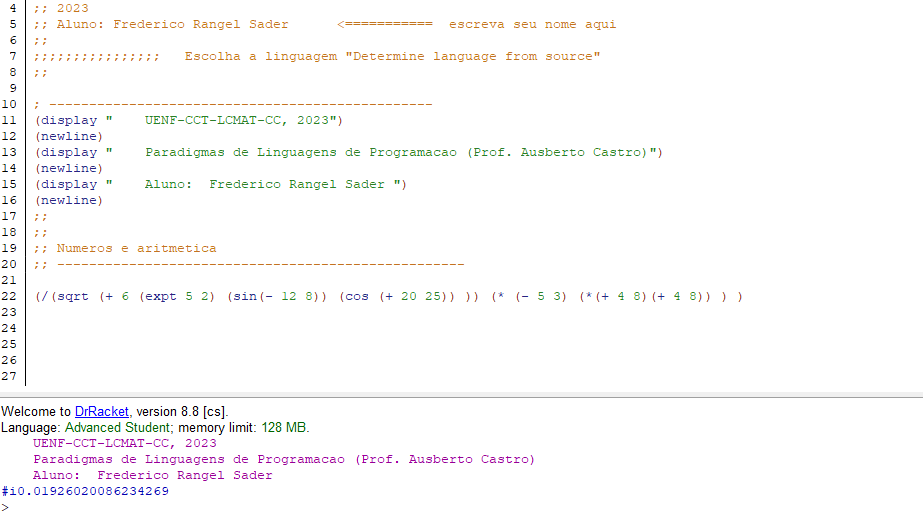


1. Escreva programas Racket para as seguintes expressões:
   1. P = (6 – (82 - 43) / 6) – (7 + (2 – (34 – 5)



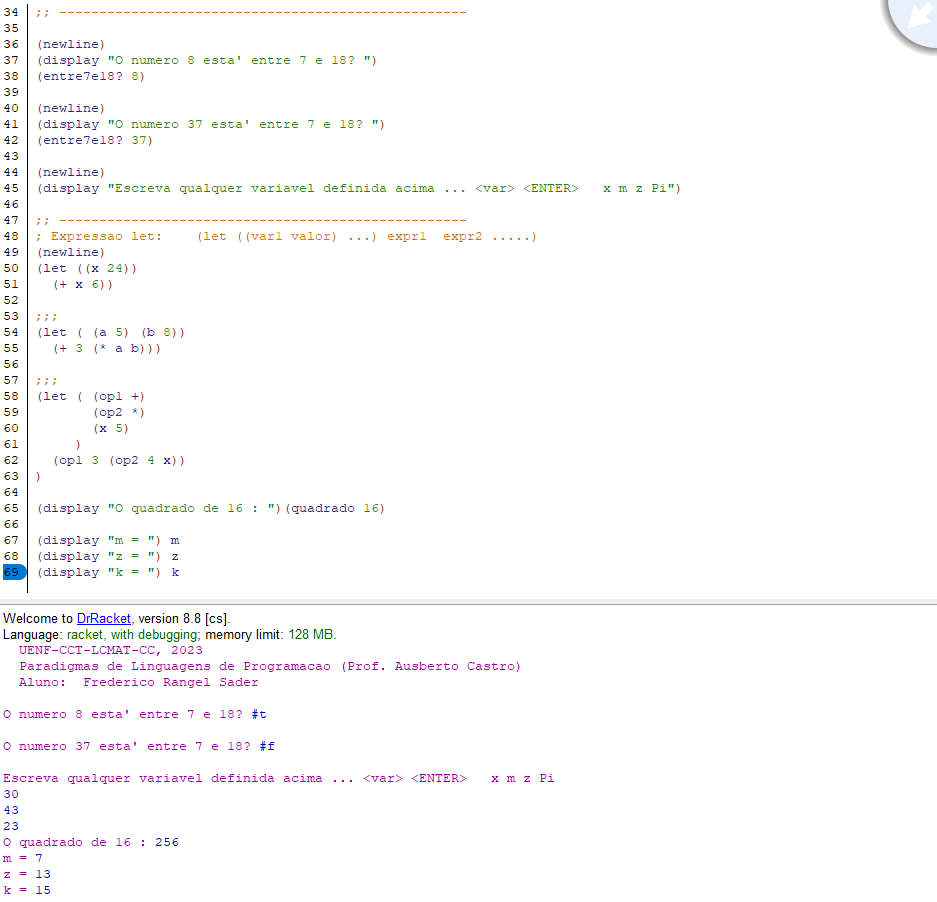
Escreva um NOVO programa Racket que calcule o valor da expressão:





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 primeira expressão let:(let ((x 24)) atribui o valor 24 à variável “x”, então faz a conta x + 6, que resulta em 30.**

**A segunda expressão let: (let ( (a 5) (b 8)), atribui o valor 5 à variável “a” e o valor 8 à variável “b”, então faz a conta 3 + (a\*b), que resulta em 43.**

**A terceira expressão let:(let ( (op1 +) (op2 \*)(x 5)), atribui a operação + à variável “op1”, a operação \* à variável “op2” e o valor 5 à variável x, então faz a conta 3 op1 (4 op2 x) , que resulta em 23.**

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

**1- (let ( (x 2) (y 6) (z 7))**

**( \*x y z))**

**Essa expressão atribui os valores 2, 6 e 7 às variáveis x,y e z, respectivamente.**

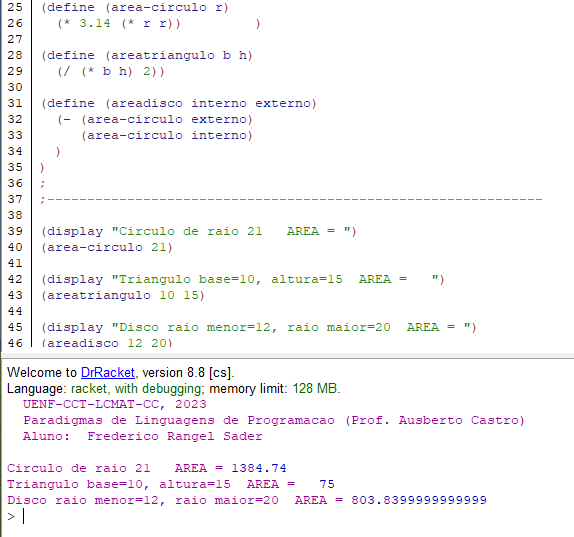
**Então faz a conta x\*y\*z, que resulta em 84.**

**2-(let ( (op1 -) (op2 +) (op3 /))**

**( op3 (op1 10 2) (op2 1 3) ) )**

**Essa expressão atribui as operações -, + e / às variáveis op1, op2 e op3, respectivamente. Então faz a conta (10 - 2)/(1 + 3), que resulta em 2.**

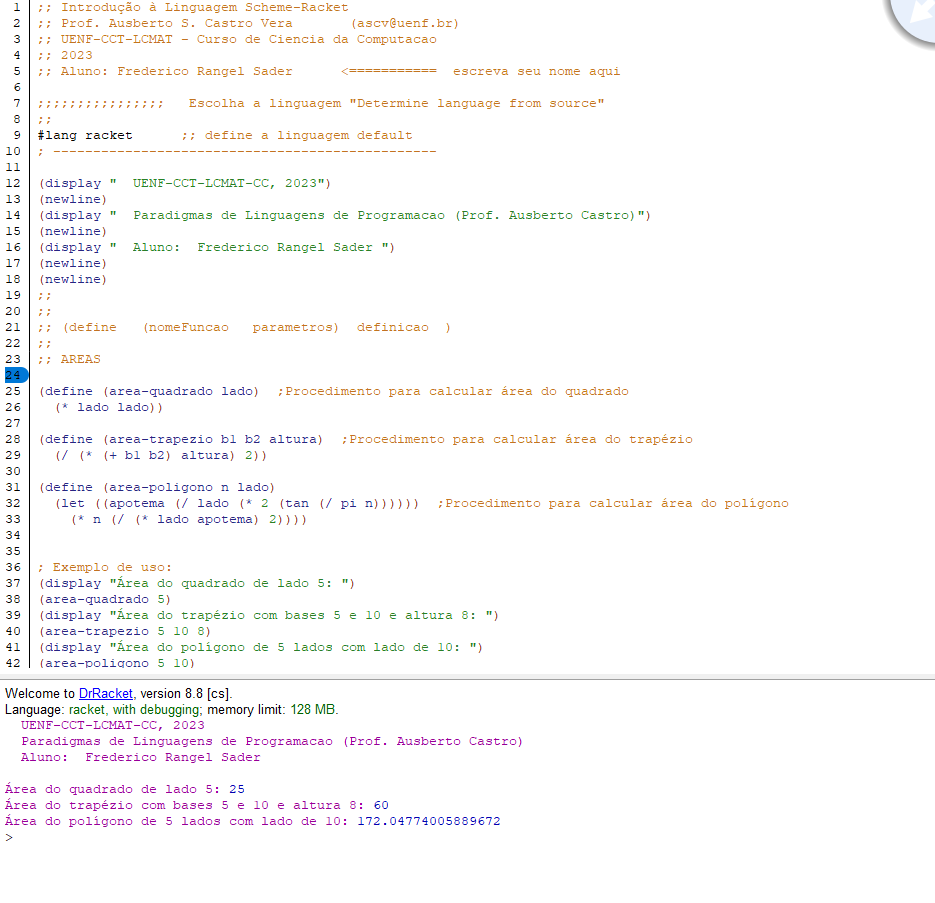
Arquivo 04-areas.rtk

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

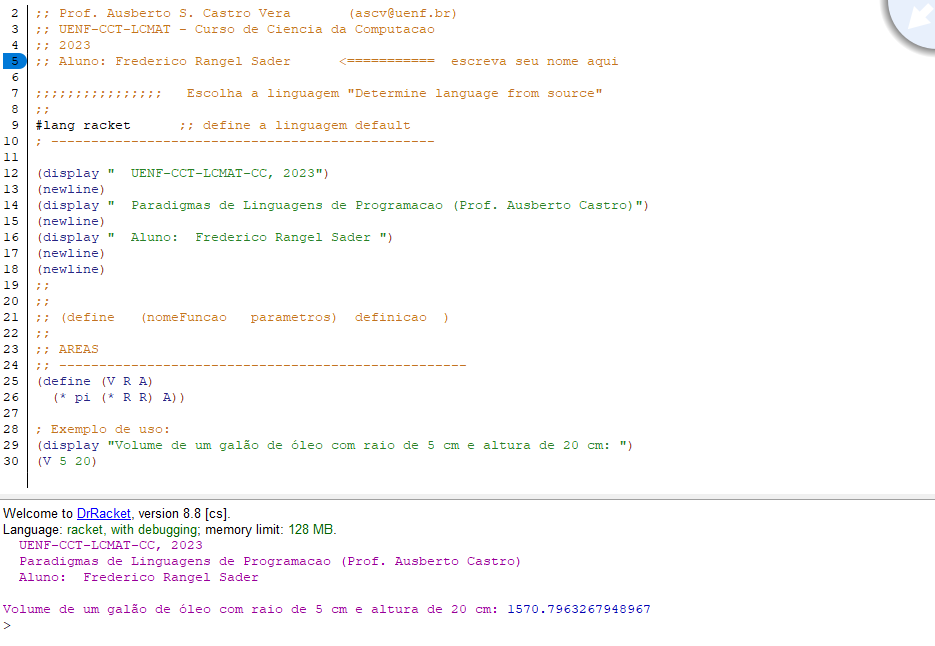
**O programa define três procedimentos: “area-circulo r”, que multiplica o quadrado da variável “r” por 3,14, “areatriangulo b h”, que divide o produto das variáveis “b” e “h” por 2, e “areadisco interno externo” que subtrai a variável "área-circulo externo” e “area-circulo interno”.**

**Depois mostra na tela os procedimentos sendo executados com o número 21 para o “area-circulo”, que resulta em 1384.74, com os números 10 e 15 para o “areatrinangulo”, resultando em 75, e os números 12 e 20 para o “areadisco interno externo”.**

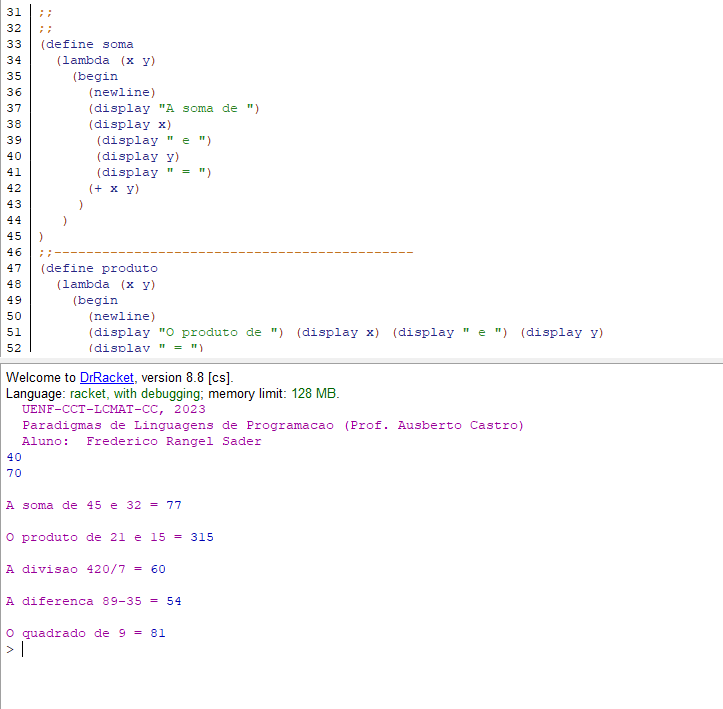
* 1. Escreva um programa Racket para calcular a área de um quadrado qualquer, a área de um trapézio e a área de um polígono.



* 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

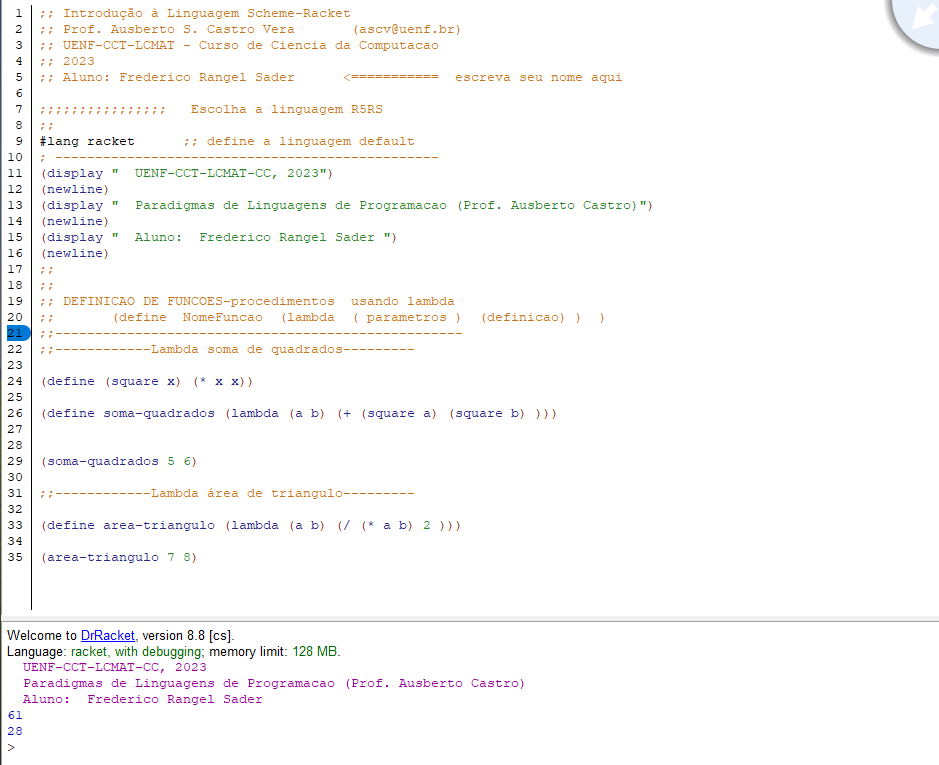


Arquivo 05-funcoes.rtk

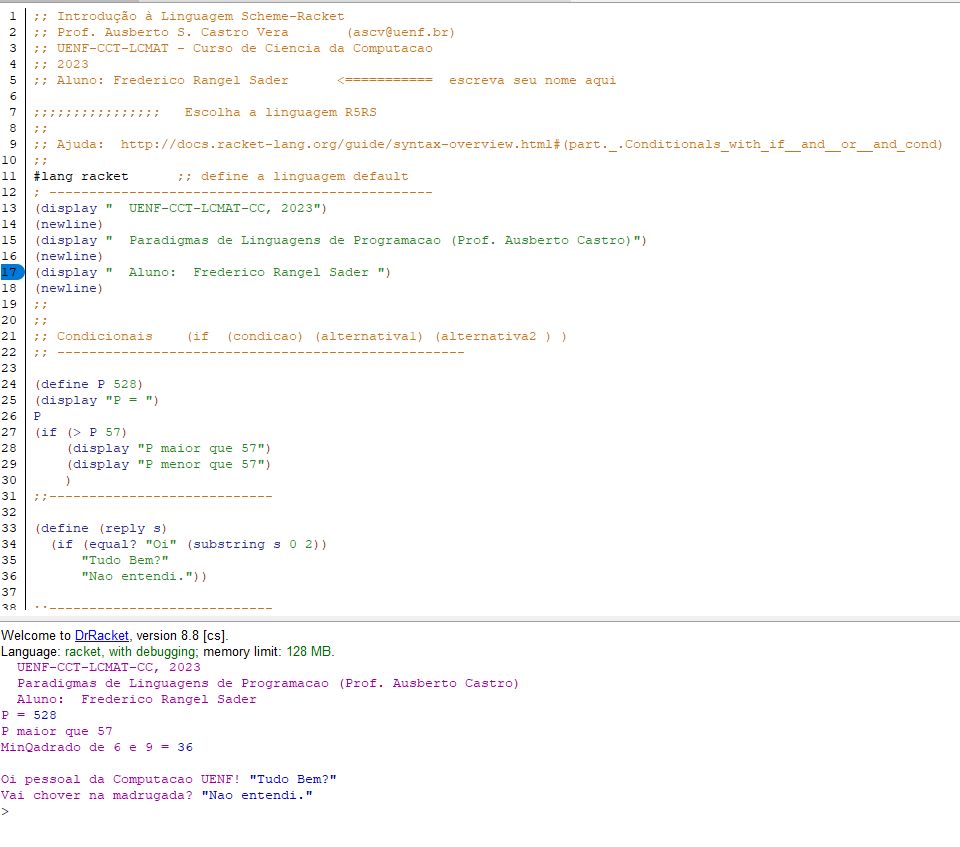
1. Execute o programa e explique o que faz

**O programa define quatro funções lambda, tendo como parâmetros x, y e a e coloca como corpo das funções a impressão das operações de soma, multiplicação, divisão, soma e potênciação.**

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



Arquivo 06-condicionalIF.rtk

1. Executar e explicar o programa

**O programa primeiro define e mostra P como 528, depois roda o condicional ‘if’, com a condição de P ser maior que 57, caso seja, imprime uma mensagem falando que é maior, caso não seja, imprime uma falando que é menor.**

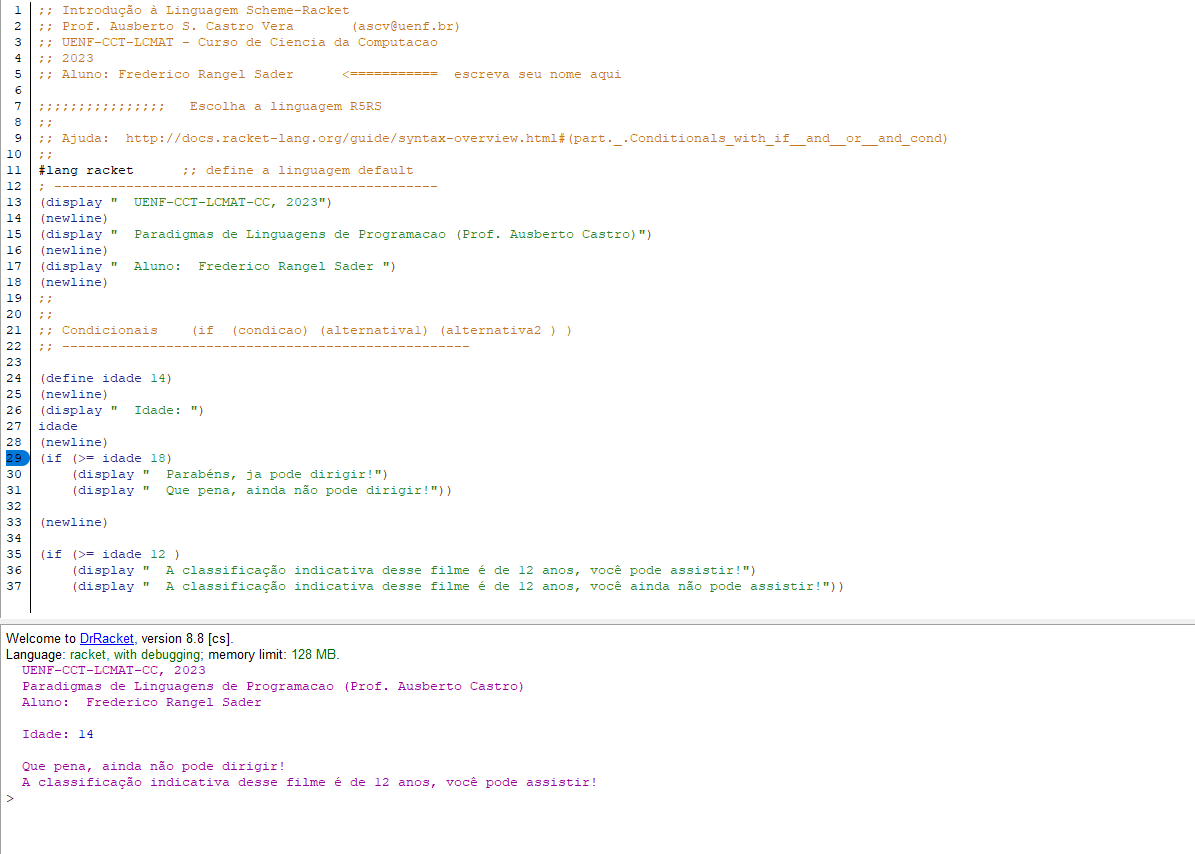
**Em seguida define um comportamento de resposta automática para uma mensagem recebida como entrada ‘s’. Se os primeiros dois caracteres da mensagem forem “Oi”, a função retorna “Tudo Bem?”. Caso contrário, a função retorna “Nao entendi.”.**

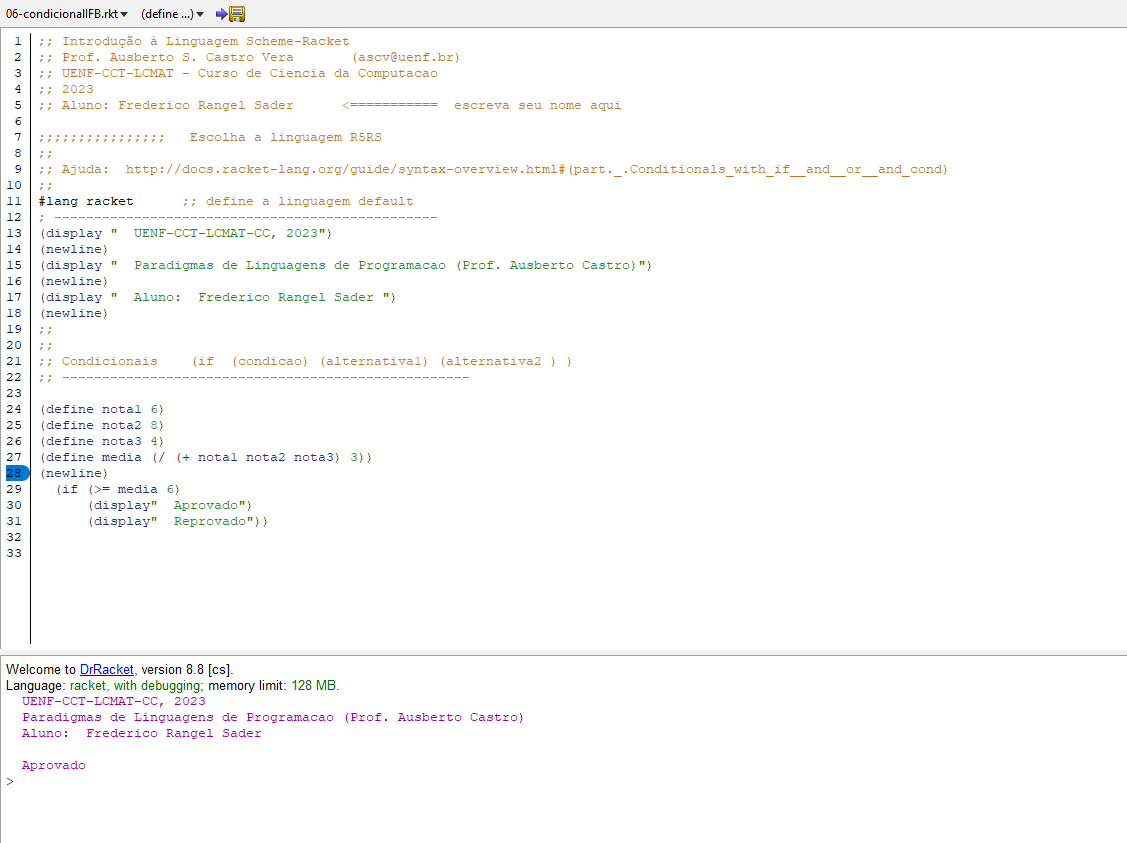
**Então define uma função chamada de quadrado que tem como parametro formal um lambda (a) que recebe a operação a\*a.**

**Então define outra função chamada de minquadrado que tem como parametros as variaveis a e b e tem um condicional ‘if’ com a condição de ‘a’ ser menor que ‘b’, se for verdade, minquadrado recebe a ‘quadrado a’ se for falso, recebe ‘quadrado b’.**

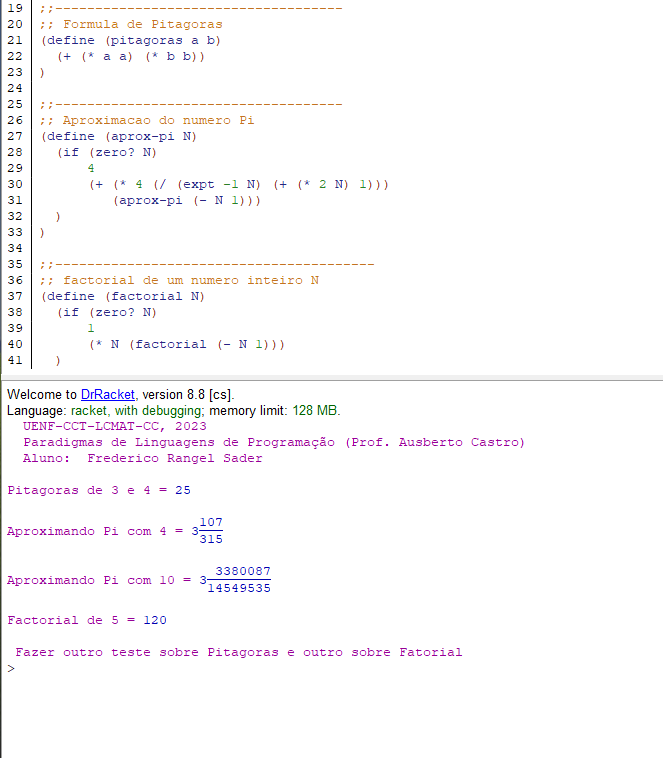
**Por fim, imprime uma mensagem fazendo o minquadrado de 6 e 9 com o resultado 36 (visto que 6 é menor que nove), e outras duas mensagens testando a função reply, uma retornando o “Tudo Bem?” e outra retornando o “Nao entendi.”.**

* 1. Escreva um programa com dois condicionais



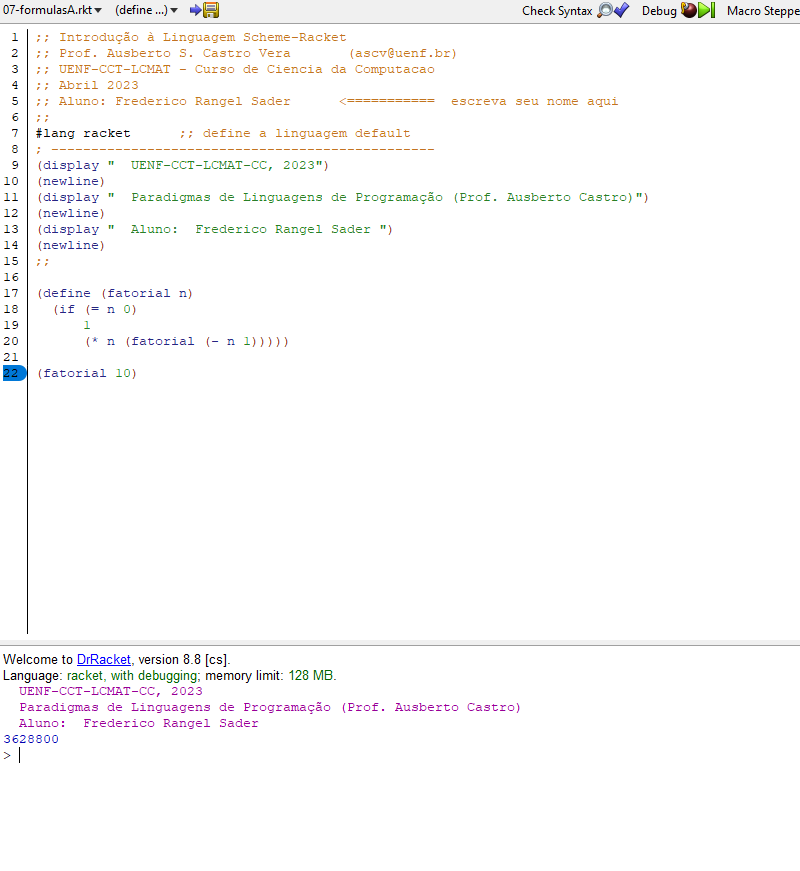
* 1. Escreva um programa para calcular a média de três notas e indique “Aprovado” se for maior ou igual a 6,0, e “Reprovado”, caso contrário 

Arquivo 07-formulas.rtk

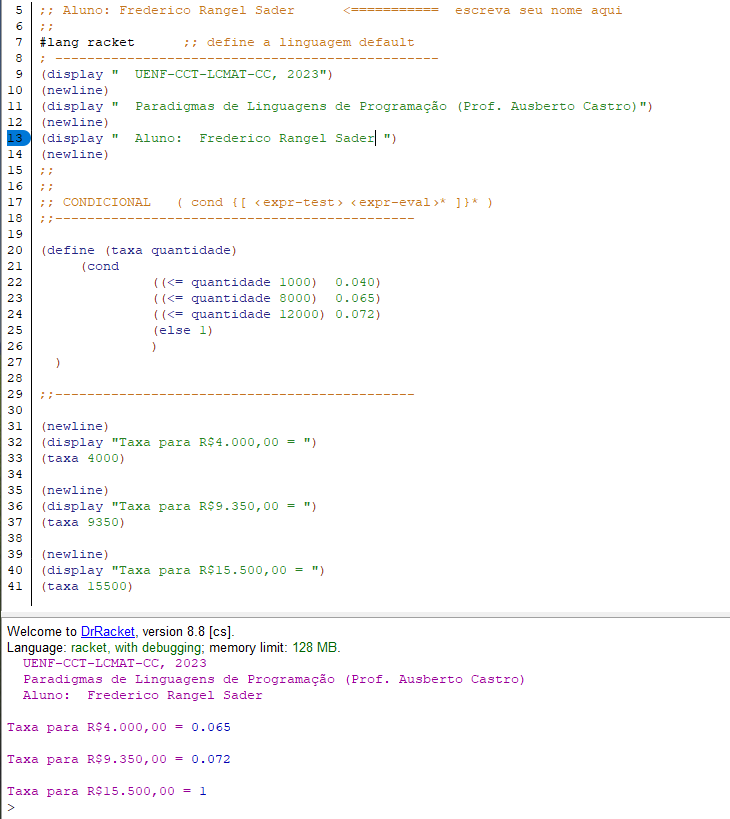
1. Executar e explicar o programa

**O programa define três funções, “pitagoras”, “aprox-pi” e “factorial n”. A primeira recebe dois argumentos ‘a’ e ‘b’, e retorna a soma dos quadrados desses valores, que é a fórmula de pitágoras.O segundo recebe um argumento n, que indica quantas iterações devem ser realizadas. Em cada iteração, a função calcula um termo da série de aproximação do número Pi e o adiciona ao resultado total. Ao final das iterações, o resultado é retornado. O último recebe um argumento ‘n’ que representa um número inteiro e retorna o fatorial desse número, que é o produto de todos os números inteiros positivos menores ou iguais a ‘n’.**

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

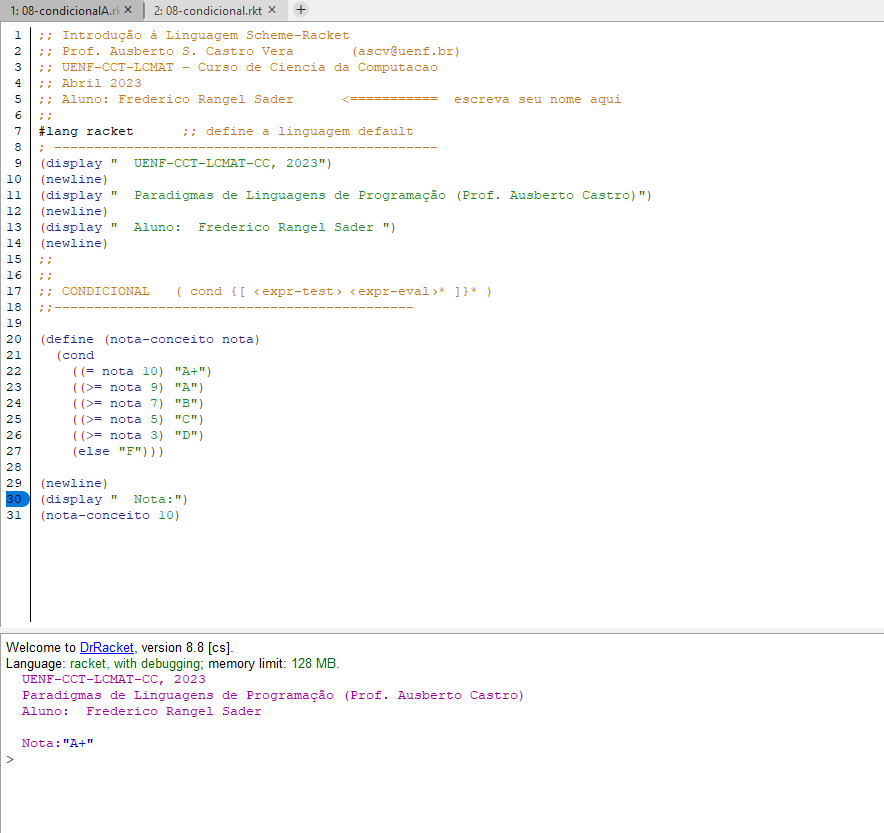


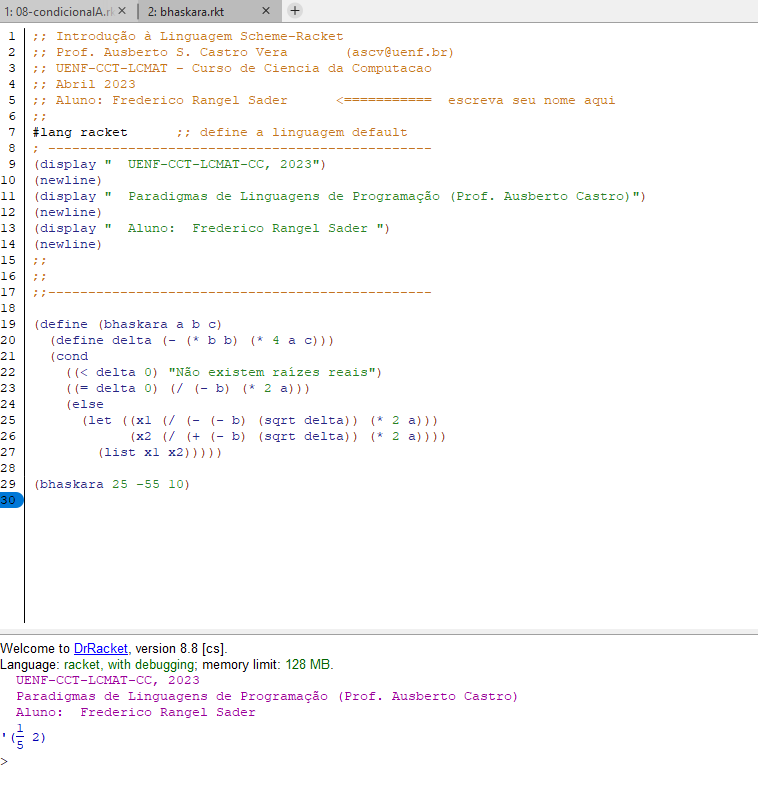
Arquivo 08-condicional.rtk

1. Execute o programa e indique o que faz

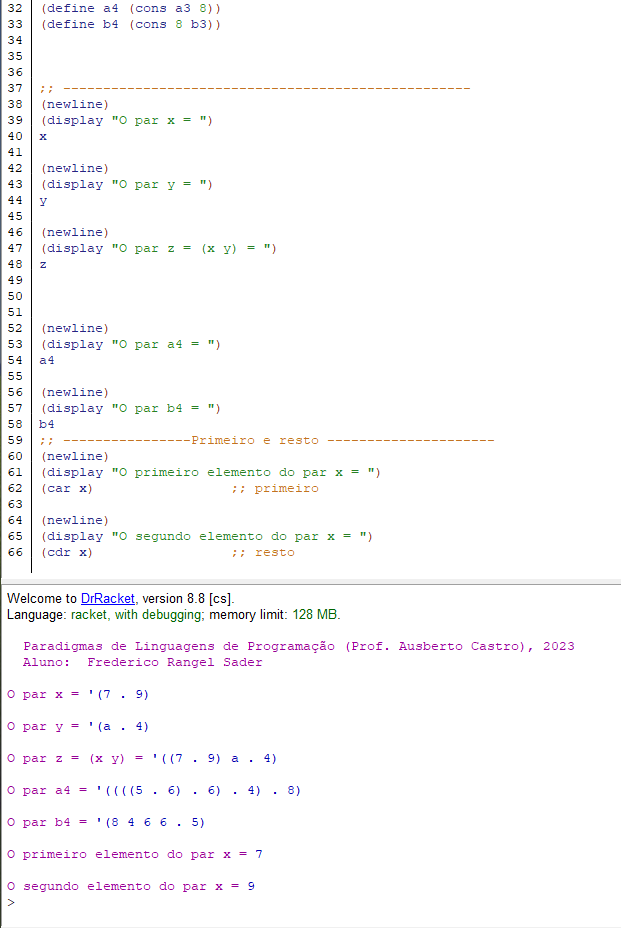
**O programa define uma função que leva como argumento a variável quantidade e roda um condicional que mostra taxas para determinados valores dessa quantidade, Então roda a função com os valores 4.000, 9.350 e 15.500.**

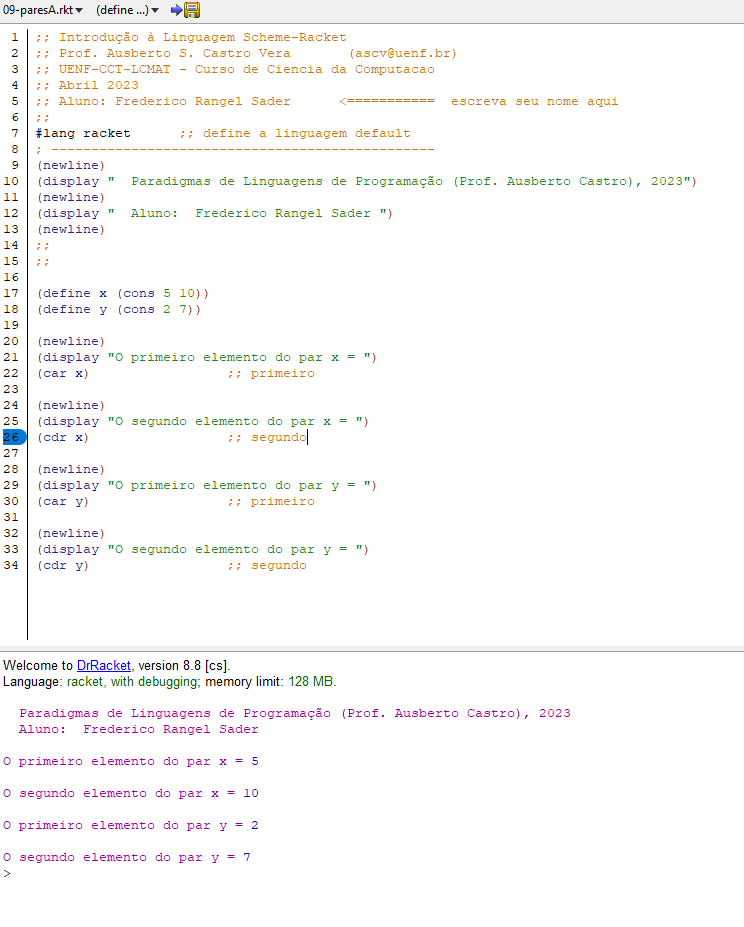
* 1. Escreva um programa condicional com pelo menos 5 opções



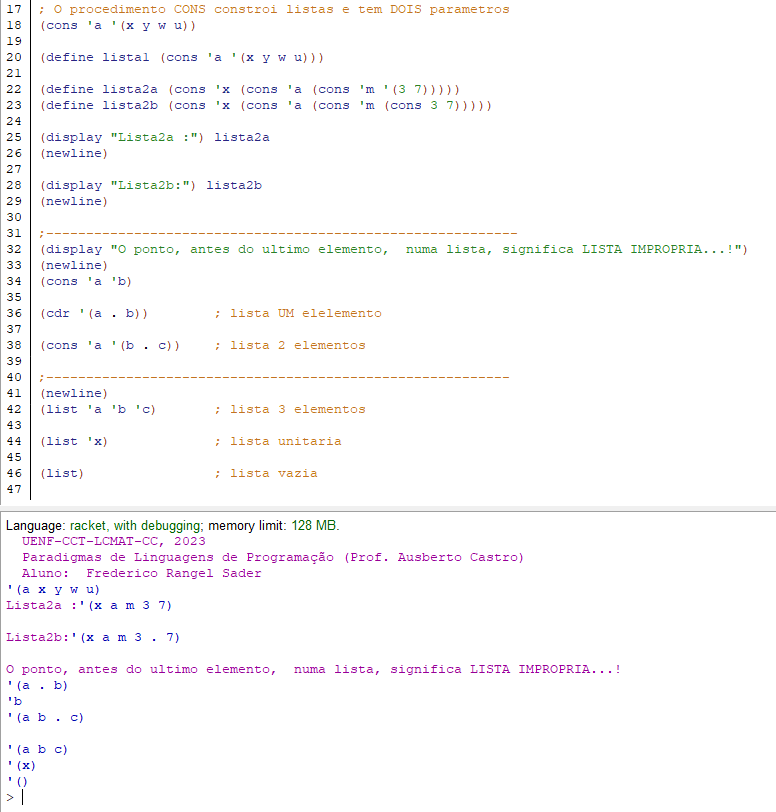
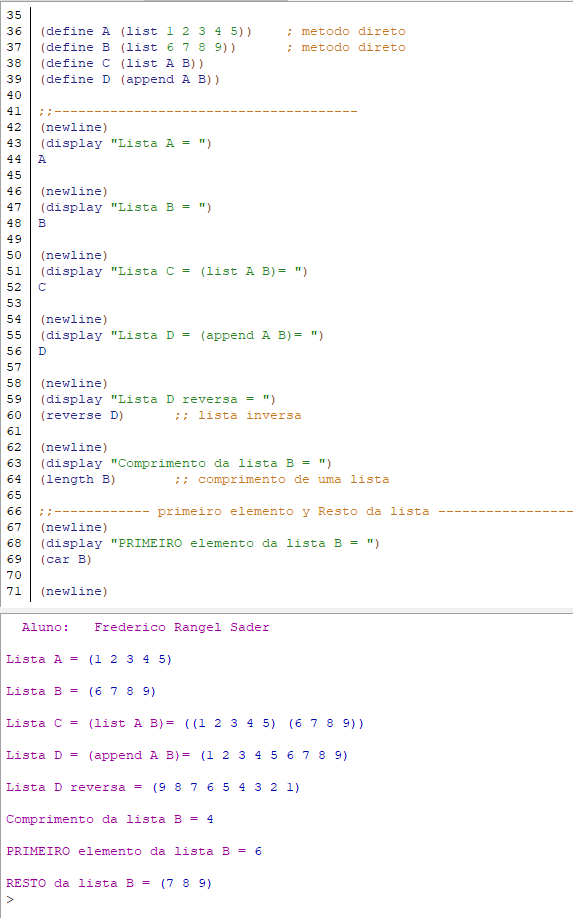
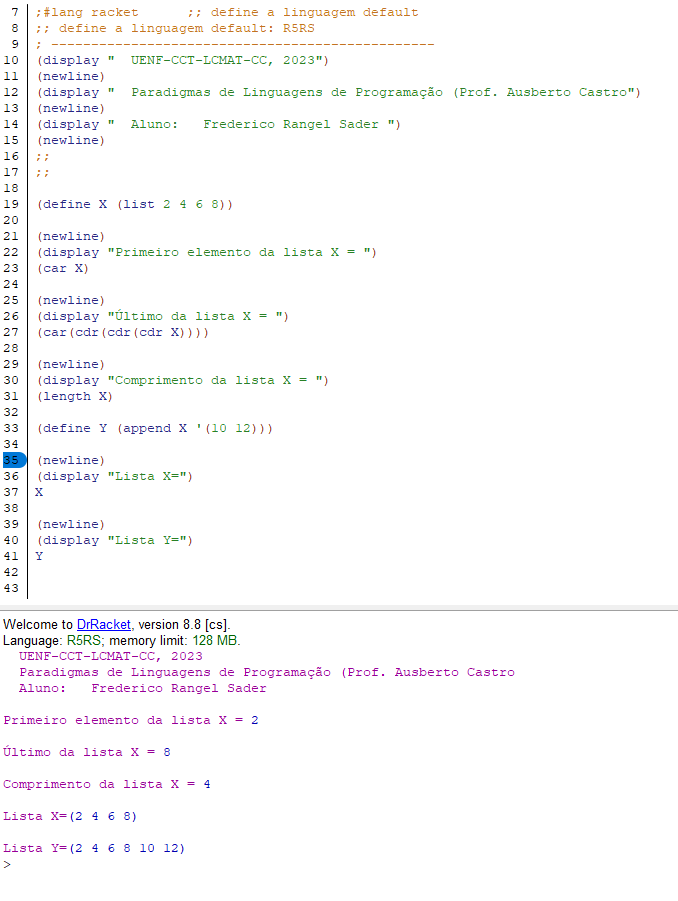
* 1. Escreva um programa bhaskara.rkt que calcule as raízes de uma equação 25x2 - 55x+10 = 0, utilizando a fórmula de Bhaskara. Sugestão: Primeiro faça o algoritmo completo 

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

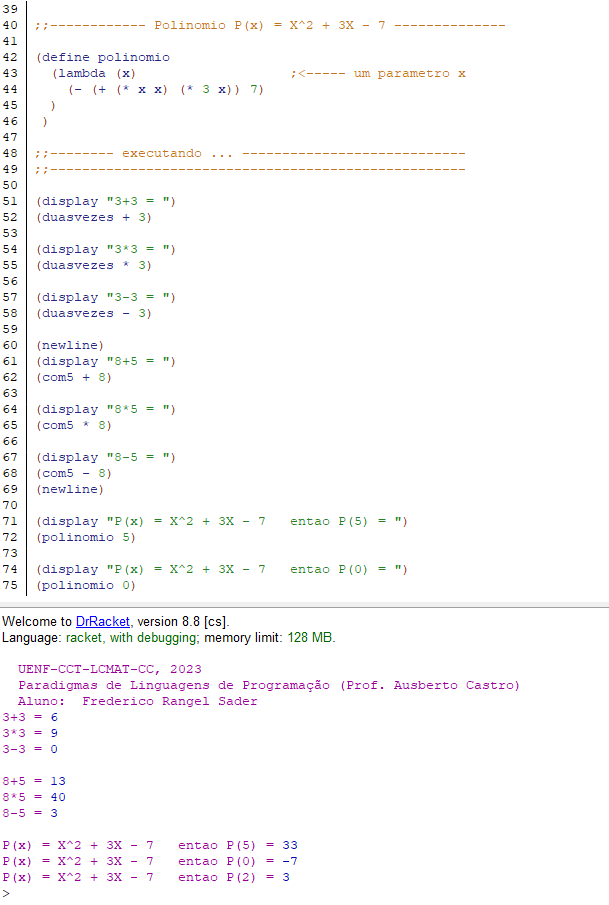


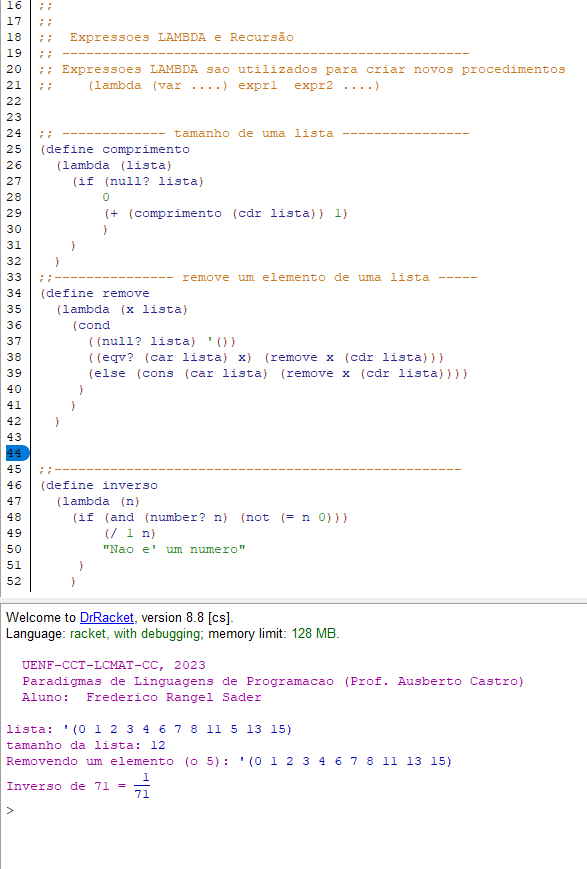
Arquivo 10-listas.rtk e 11-listas.rtk

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
   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)

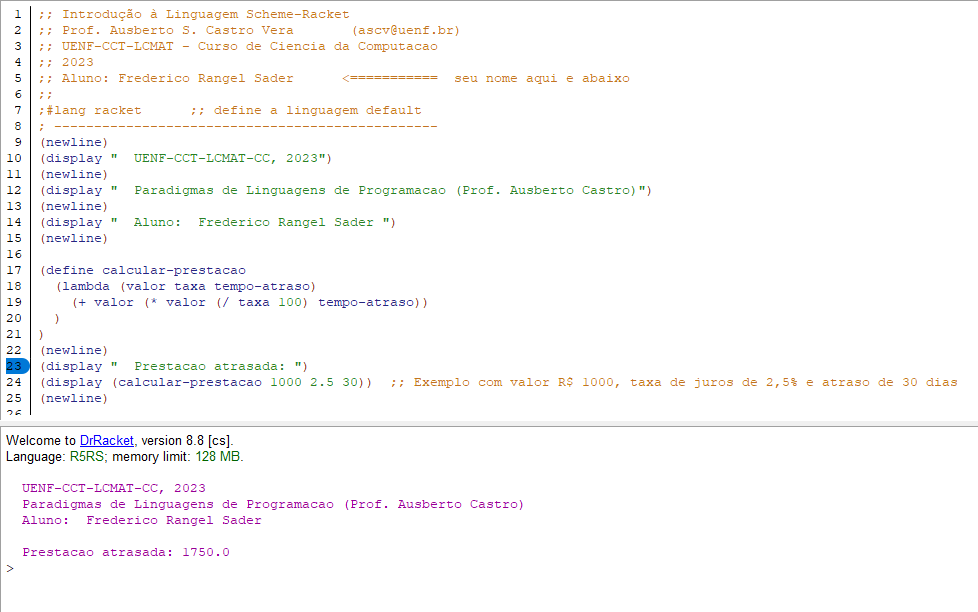
Arquivo 12-lambda.rtk e 13-lambda.rtk

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



**O código define três funções, ‘duasvezes’, ‘com5’ e ‘polinomio’, todas utilizando lambda. A primeira recebe duas entradas, uma função ‘f’ e um valor ‘x’, e aplica ‘f’ em ‘x’ duas vezes. Por exemplo, ‘duasvezes + 3’ retorna o valor ‘9’, que é ‘(+3 3)’. A segunda recebe duas entradas, uma operação aritmética e um valor ‘x’, e aplica a operação em ‘x’ e ‘5’. Por exemplo, ‘com5 + 8’ retorna o valor ‘13’, que é ‘(+ 8 5)’. A última recebe um valor ‘x’ e retorna o resultado do polinômio ‘ P(x) = x^2 + 3x - 7’. Por exemplo, ‘polinomio 5’ retorna o valor ‘33’.**

**O código define três funções com lambda, ‘comprimento’, ‘remove’ e ‘inverso’. A primeira recebe uma lista como parâmetro e retorna seu tamanho, a segunda recebe um elemento ‘x’ e uma lista e retorna a lista sem o elemento ‘x’, a terceira recebe um número ‘n’ e retorna o seu inverso. Se ‘n’ não for um número ou se for zero, retorna a mensagem “Nao e um número”.**

* 1. Crie um procedimento para realizar o cálculo de uma prestação em atraso, utilizando a fórmula Prest = valor + (valor\*(taxa/100)\*tempo). Dar exemplos.
  2. O que faz o seguinte procedimento **abcd**:

(define abcd

(lambda (n)

(let f ((i 2))

(cond

((>= i n) '())

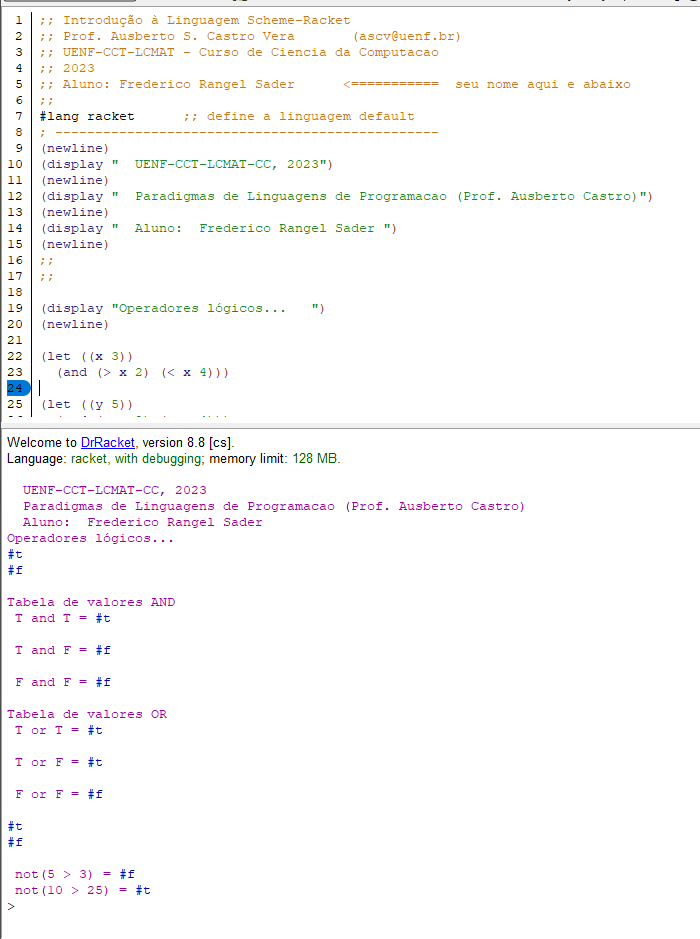
((integer? (/ n i))

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

(else (f (+ i 1)))))))

**O procedimento recebe um argumento ‘n’ e retorna uma lista dos fatores inteiros de ‘n’ maiores que 1. Define um procedimento interno com o ‘let’ chamado de f, que recebe um argumento 'i'. Dentro do procedimento ‘f’, a primeira condição é a de se i é maior ou igual a ‘n’, e se for, retorna uma lista vazia. A segunda condição é a de se ‘n’ é divisível por ‘i’. Se for, ‘i’ é adicionado à lista resultante e o procedimento ‘f’ é chamado recursivamente com’i’ incrementado em 1. Se não for, i não é adicionado à lista e o f é chamado recursivamente mais uma vez.**

Arquivo 14-operad-logicos.rtk Operadores lógicos

1. Executar o programa e indicar o seu conteúdo

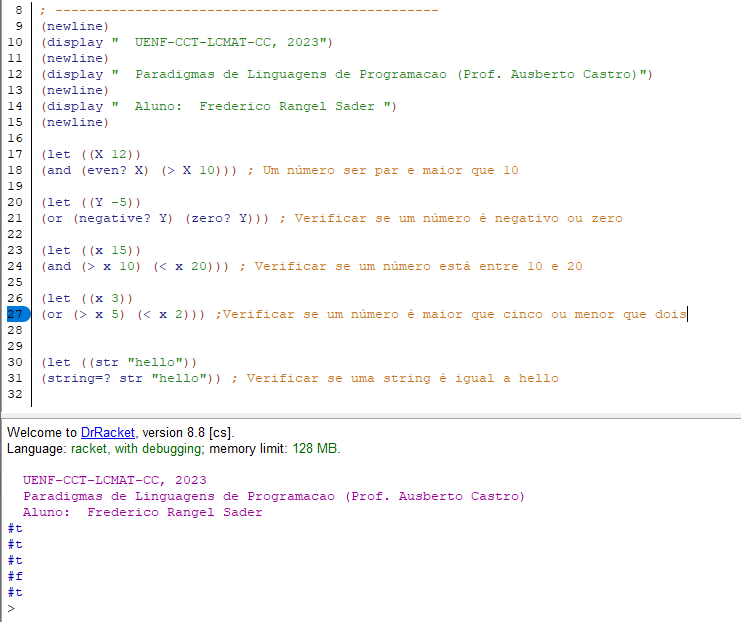
**O programa utiliza o operador lógico ‘AND’ para verificar se o número três é maior que 2 e menor que 4, em seguida, faz o mesmo com o número 5.**

**Depois, demonstra como funciona o operador ‘AND’ de uma forma prática, imprimindo sua tabela verdade que mostra que o resultado só sera verdadeiro se ambas as expressões forem verdadeiras. Faz o mesmo com o operador ‘OR’, mostrando que o resultado será verdadeiro se pelo menos um das expressões for verdadeira.**

**Então usa o operador ‘or’ para verificar se o número 3 é maior que 2 ou menor que 8, e se o número 10 é maior que 20 ou menor que 7. Ambas retornando negativo**.

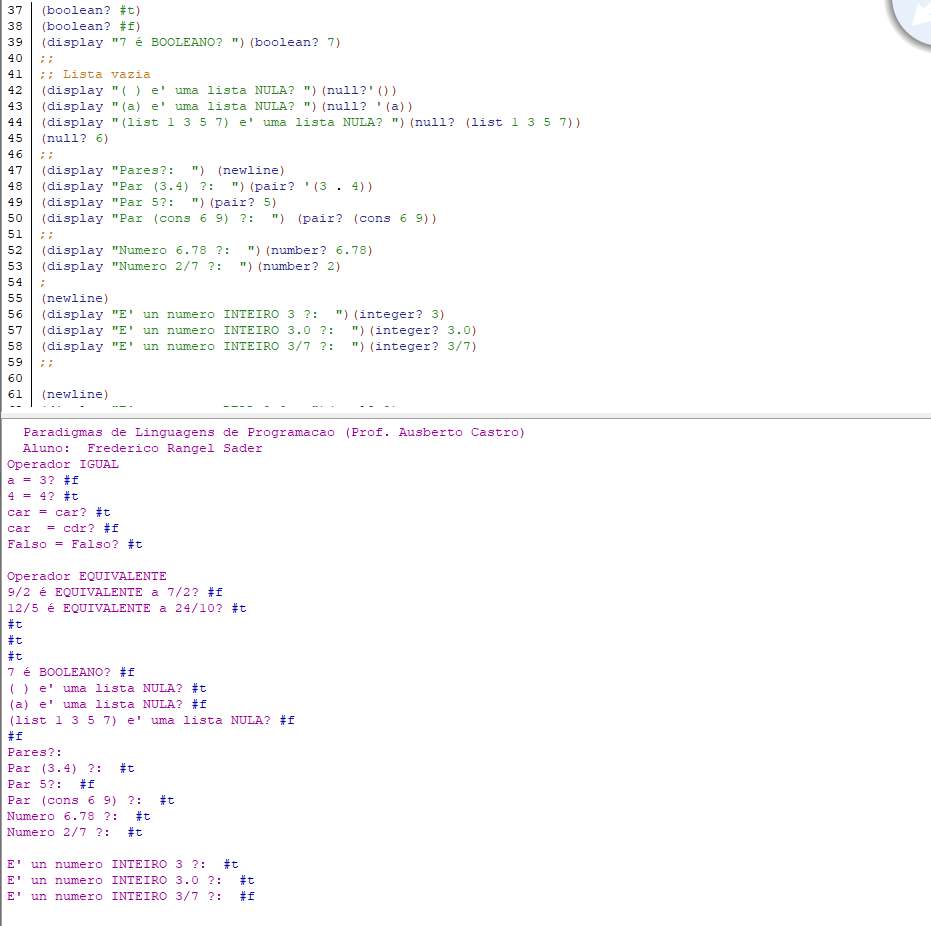
**Por fim, utiliza o operador ‘not’ para negar o resultado das expressões ‘5 > 3’ e ‘10 > 25’.**

* 1. Escreva e teste pelo menos cinco operações lógicas

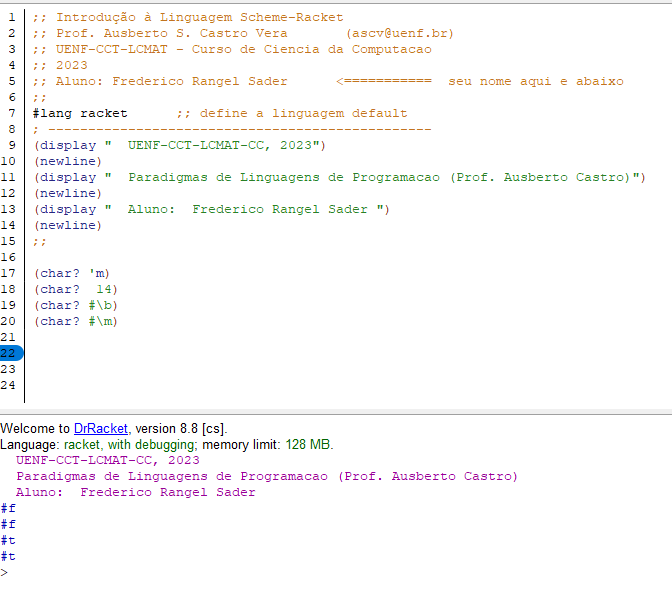


Arquivo 15-predicados.rtk Predicados

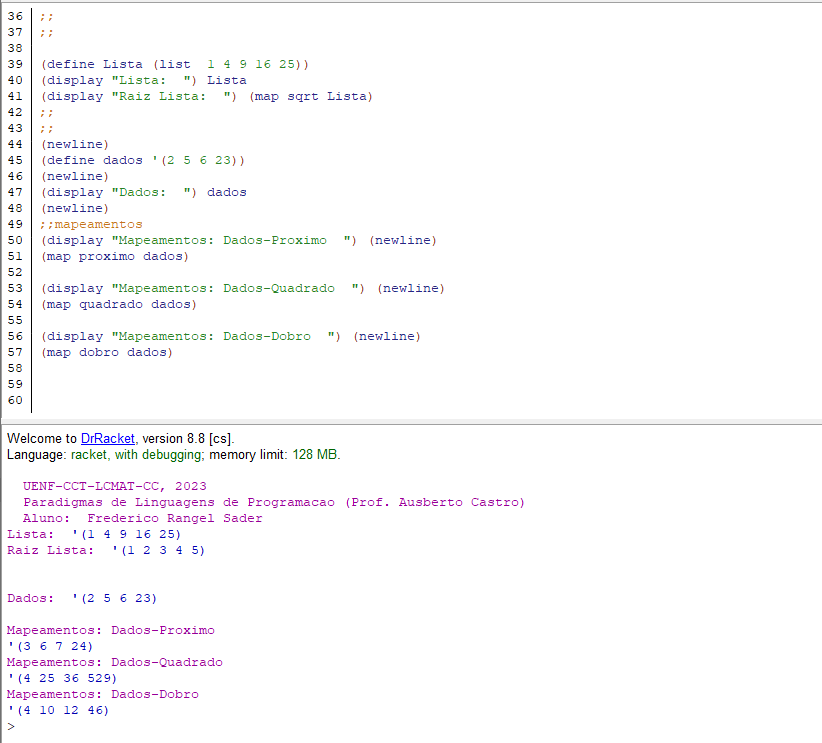
1. Executar o programa e indicar o seu conteúdo



**O programa testa alguns tipos de predicados, que são funções que retornam valores booleanos. Os predicados utilizados foram o ‘eq?’ que testa se dois argumentos são iguais, o ‘eqv?’ se são equivalentes, o ‘boolean?’ se um argumento é booleano, o ‘null?’ se uma lista é vazia, o ‘pair’ se argumentos são pares, o ’number?’ se um argumento é número, o ‘integer’ se um argumento é inteiro, o ‘real’ se é real, o ‘complex?’ se é complexo, e por fim, o ‘string?’ se um argumento é string.**

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

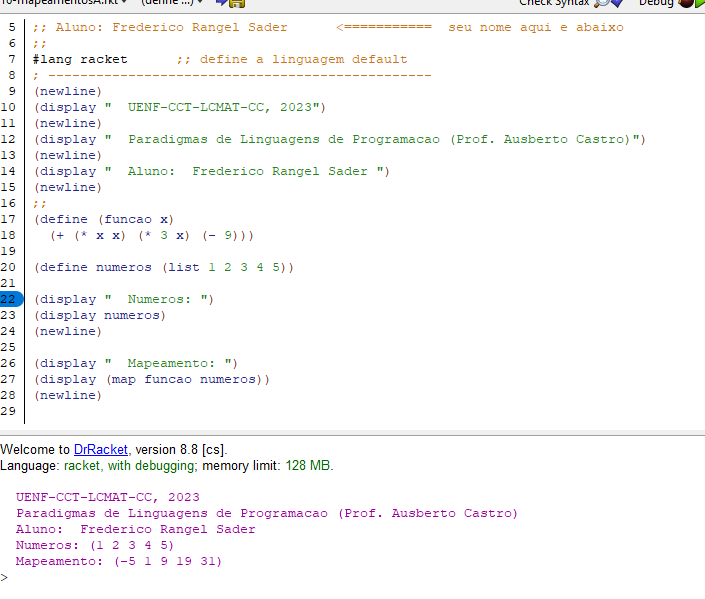
Arquivo 16-mapeamentos.rtk Mapeamentos

* 1. Executar o programa e indicar o que ele faz

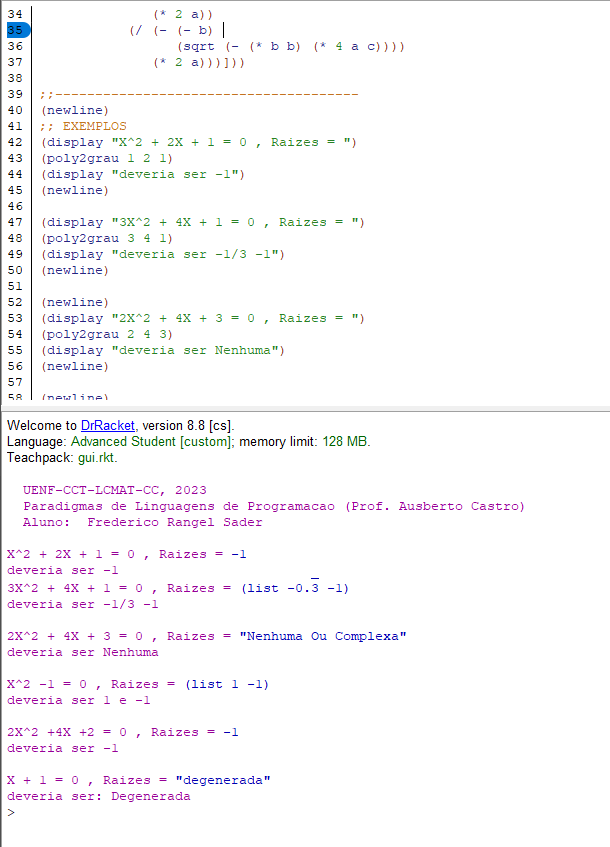
**O programa define funções para o quadrado do argumento, para o dobro do argumento e para o próximo número do argumento. Depois cria uma lista e utiliza a função ‘map’ que utiliza a função sqrt na lista criada, retornando a raiz quadrada dos números da lista. Então utiliza o mapeamento novamente para criar uma lista com os próximos números respectivamente da antiga lista, do quadrado dos números da antiga lista e do dobro dos números da antiga lista.**

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

x ⎯→ x2 + 3x - 9

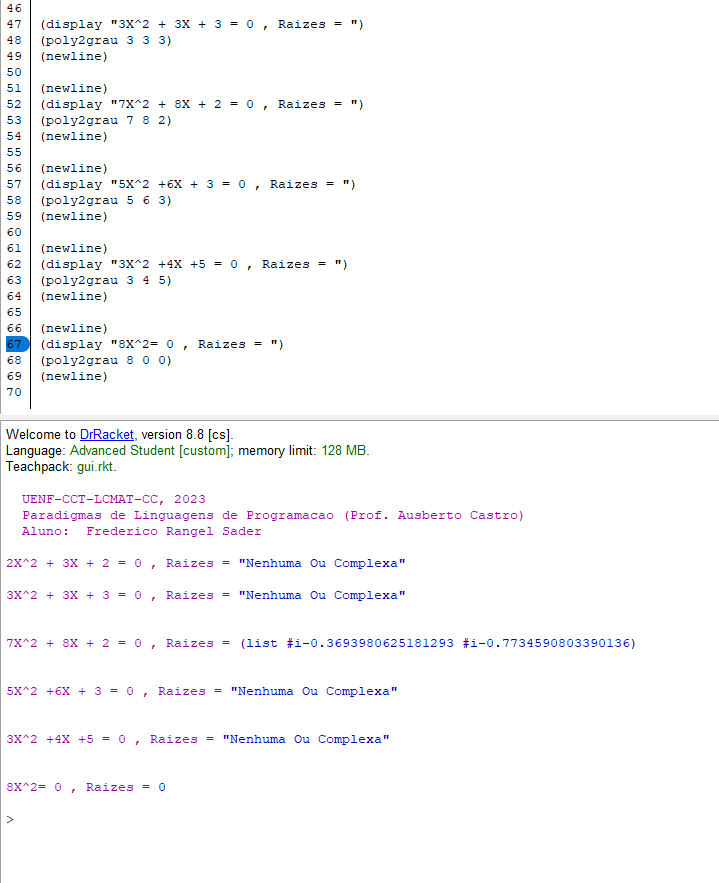


Arquivo 17-raizes-poly.rtk Aplicações: Raízes de polinômios

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

**O programa implementa uma função ‘ploy2grau’ que recebe os coeficientes de uma equação do segundo grau e utiliza ‘cond’ para verificar as condições necessárias. Primeiro verifica se a equação é degenerada, depois se a equação não tem raízes reais, e a terceira, se a equação tem uma única raiz real, retornando esse valor. Por fim, verifica se tem duas raízes, em caso positivo, calcula com a fórmula de bhaskara e retorna o resultado das duas raizes.**

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



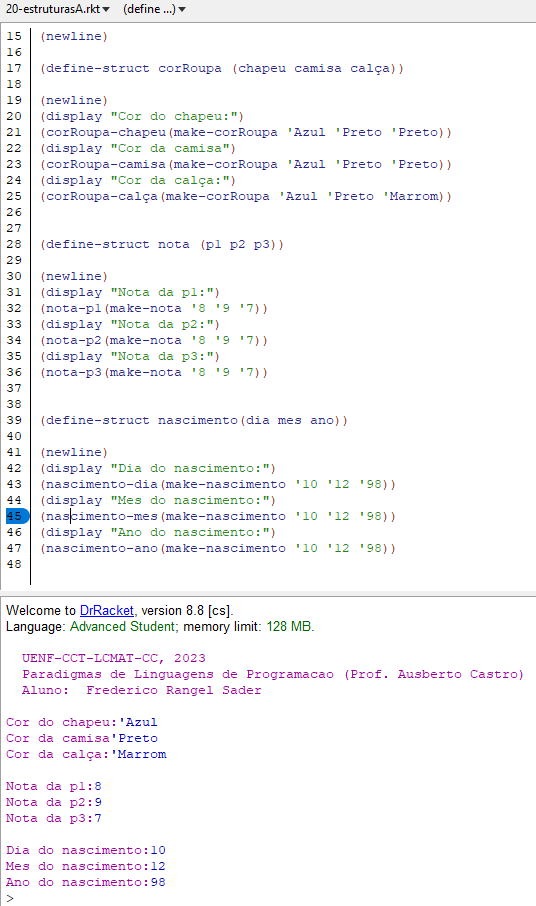
Arquivo 20-estruturas.rtk Aplicações: Estruturas de dados

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



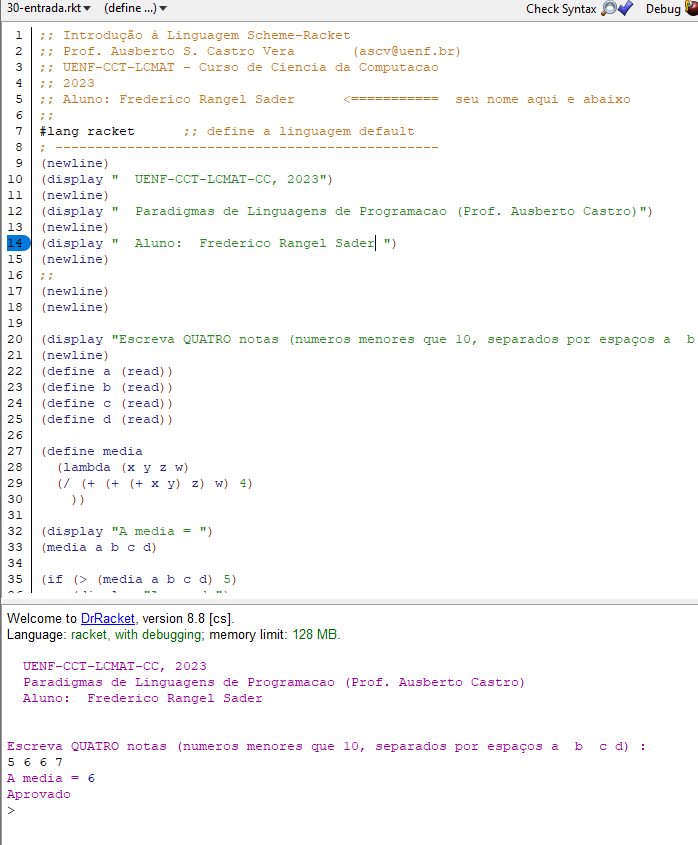
**O programa utiliza a estrutura ‘struct’ para criar uma nova estrutura chamada entrada com três dados de um usuário, nome, cep e telefone. E então imprime os três.**

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



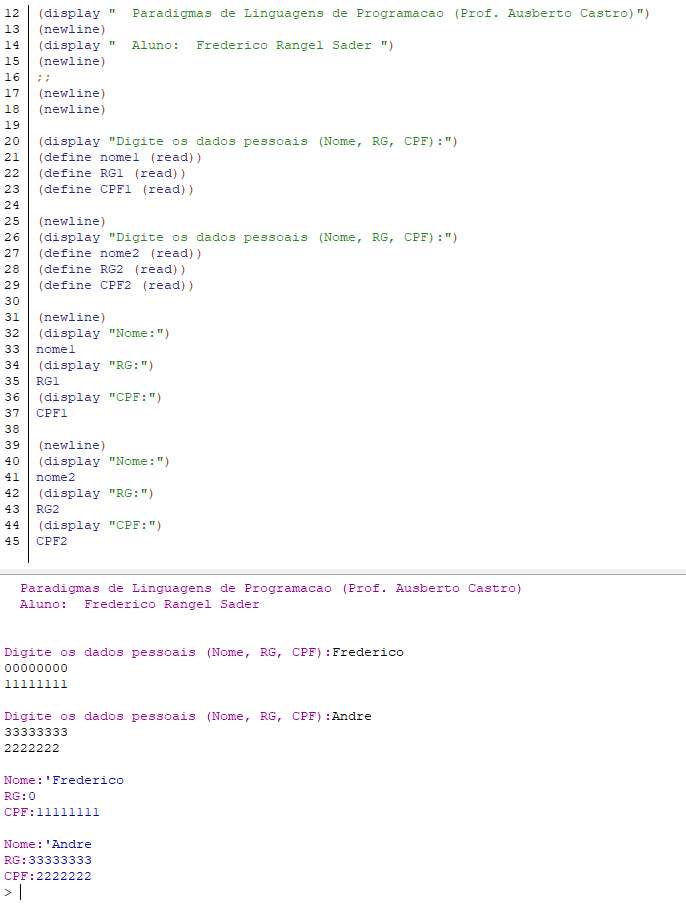
Arquivo 30-entrada.rtk

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

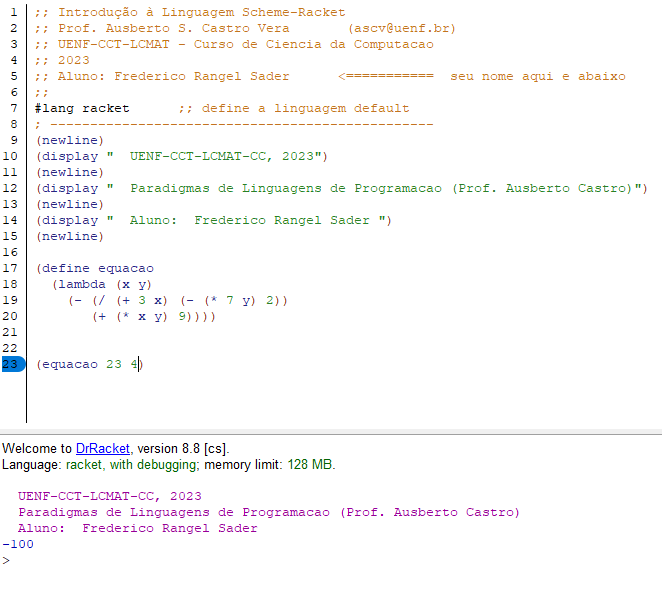
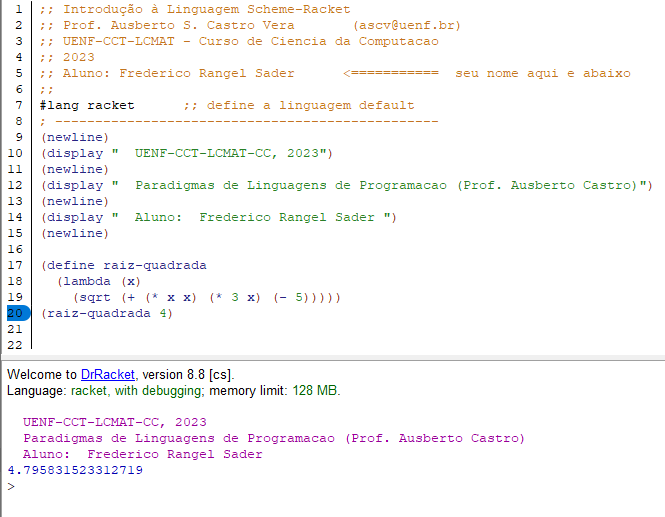
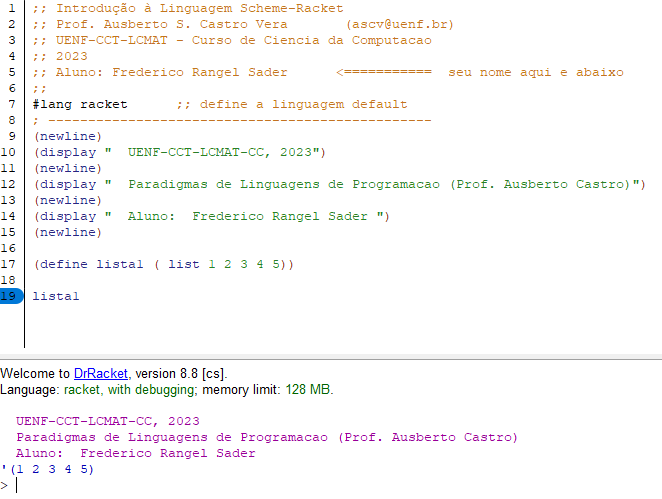
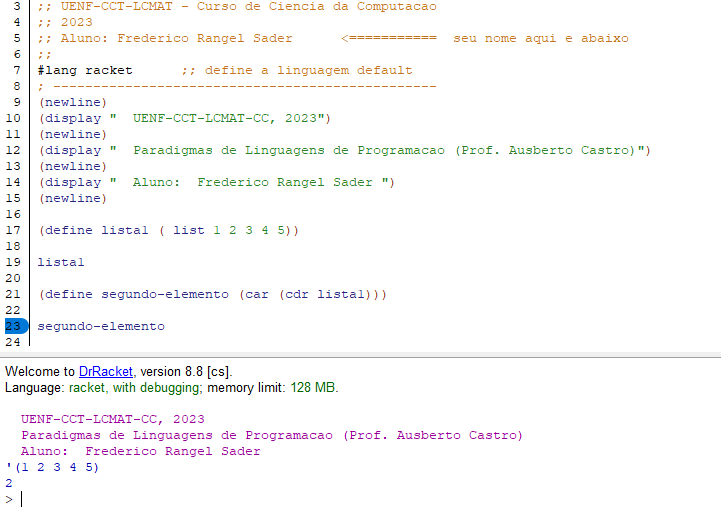
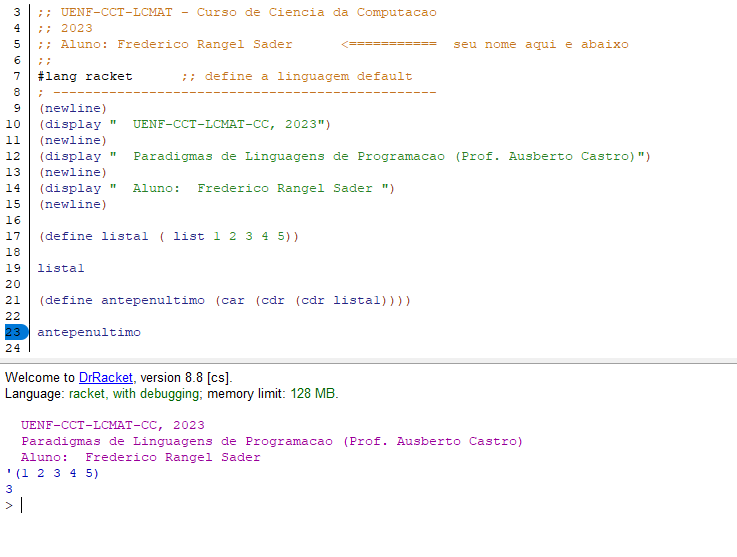
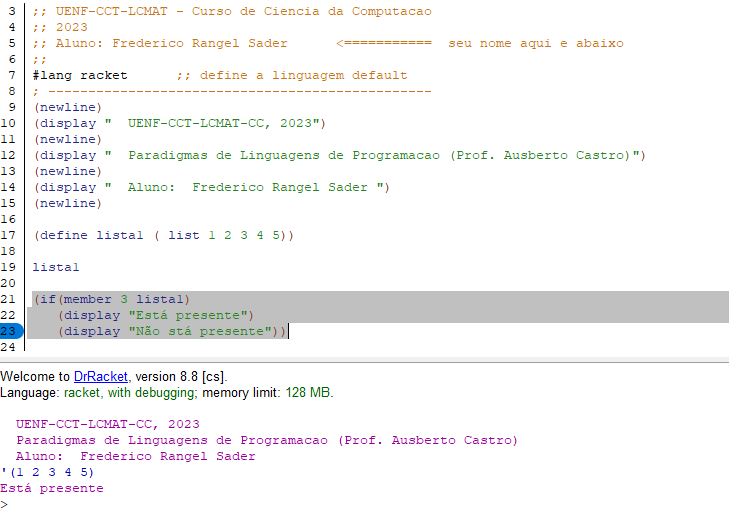
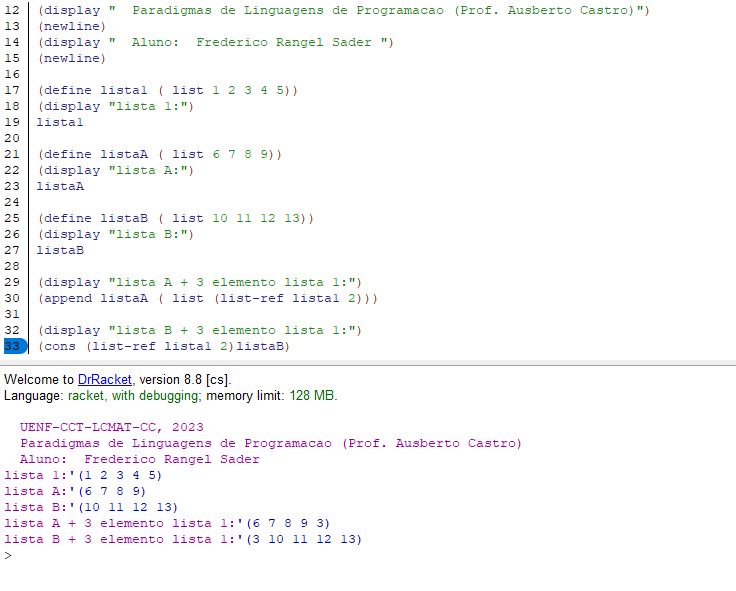
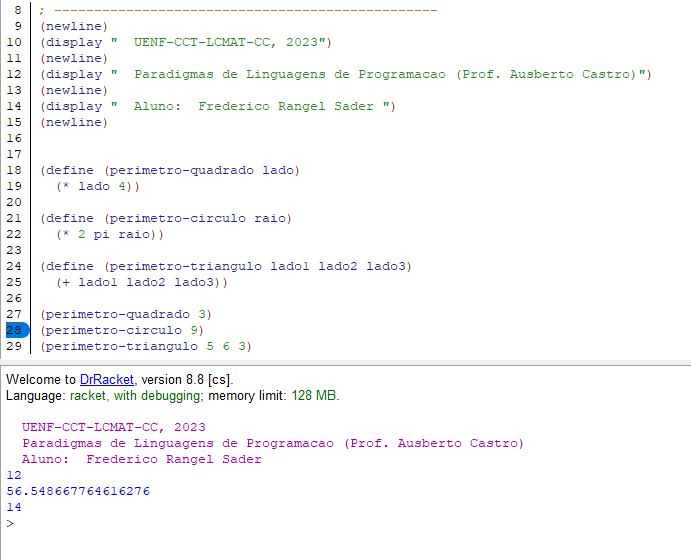
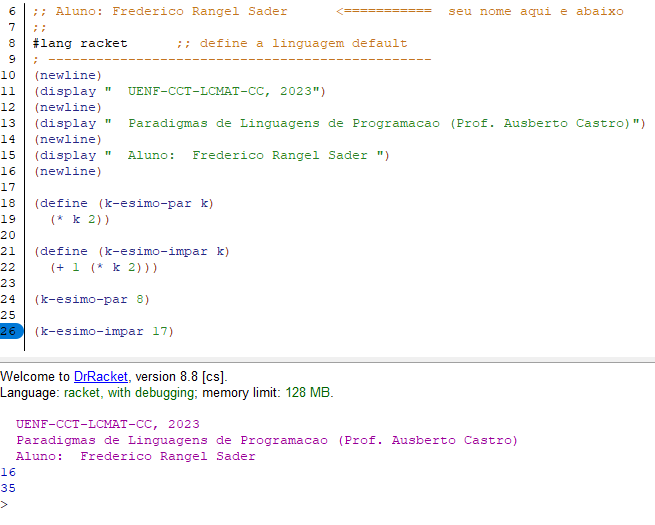


**O programa utiliza a função ‘read’ para definir quatro variáveis se baseando na entrada do usuário, então faz a média desses quatro valores e confere se é maior que 6 (aprovado) ou menor (reprovado).**

* 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)**