

UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA
CIÊNCIA DA COMPUTAÇÃO
PROGRAMAÇÃO DECLARATIVA
STÊNIO LONGO ARAÚJO

ATIVIDADE DE PROLOG 02

COMPONENTES DO GRUPO:

- ALAN BOMFIM
- BRENO CARVALHO
- DENISE NOGUEIRA
- ÉRIC VINÍCIUS
- VITOR ROSENBERGRE

1- Questão

% Base de dados

% Fatos

% FUNCIONARIO(MATRICULA,NOME,SALARIO,DATA_DE_ADMISSAO).
dataDeAdmissao(dia,mes,ano).

funcionario(201912192,vitor,2000,dataDeAdmissao(05,10,2019)).
funcionario(201711873,pedro,1000,dataDeAdmissao(04,10,2017)).
funcionario(201911543,vinicius,1500,dataDeAdmissao(06,10,2015)).
funcionario(201512721,alvaro,900,dataDeAdmissao(03,10,2019)).
funcionario(201811982,clementina,3000,dataDeAdmissao(07,10,2018)).
funcionario(201412548,claudia,2400,dataDeAdmissao(08,10,2014)).
funcionario(201711275,gustavo,1200,dataDeAdmissao(02,10,2017)).
funcionario(201912432,leonardo,2700,dataDeAdmissao(09,10,2019)).
funcionario(201912375,alessandro,500,dataDeAdmissao(10,10,2019)).
funcionario(201911311,erica,1300,dataDeAdmissao(11,10,2019)).

% Regras

% conclusao :- condicoes

% a questao vai pegar todos os funcionarios, verificar qual deles possui o salario % entre 1000 e 2000 e vai imprimir o nome daqueles que estiverem dentro das condições.

questao01a(x) :- funcionario(_,Nome,Salario,_), Salario >= 1000, Salario <=2000,
write("Nome: "),write(Nome),nl.

% a questao utiliza dos dados do funcionario, para verificar se o ano é igual ou maior que % 2018 e se o salario ganho é maior que 2000. Se isso for verdadeiro, então imprime o % nome e a matricula da pessoa. questao01b(x) :-
funcionario(Matricula,Nome,Salario,dataDeAdmissao(_,_,Ano)), Ano > 2018, Salario >2000, write("Matricula: "), write(Matricula), nl, write("Nome: "), write(Nome),nl.

```
%regra que testa a primeira questão
test_1(x) :-
    trace, questao01a(x);
    trace, questao01b(x).
```

2- Questão

```
% Fatos
```

```
% representacao do grafo atraves de suas arestas (estradas nesse caso)
% cada estrada tem uma cidade de origem, uma de chegada e a distancia entre elas
% estrada(Origem Destino, Km)
estrada(a,d,23).
estrada(a,b,25).
estrada(b,c,19).
estrada(b,e,32).
estrada(c,d,14). estrada(c,f,28).
estrada(d,f,30). estrada(e,f,26).
```

```
% Regras
```

```
% dist eh a relação transitiva que calcula a distancia entre uma cidade A e uma B
% caso base (de A se chega diretamente na em B, são adjacentes)
dist(A,B,D) :- estrada(A,B,D).
% caso geral da cidade A se chega na T com uma distancia D1 e de T calculamos
% a distancia D2 para B, ao final calculamos a distancia D como a soma das
% duas outras distancias dist(A,B,D) :- estrada(A,T,D1),
dist(T,B,D2), D is D1 + D2.
```

```
% teste teste(X) :trace, estrada(a,d,23), estrada(b,c,D), write(D),
estrada(b,e,32); trace, dist(b,d,D), write(D), dist(b,a,D1).
```

3-11 Questões4

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% 3. numero_par(X) /* X é um número natural par */
numero_par(N) :- N mod 2 == 0.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% 4. numero_impar(X) /* X é um número natural ímpar */
numero_impar(N) :- N mod 2 == 1.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% 5. fibonacci(N,R) /* R é o número da série de Fibonacci na posição N */
fibonacci(0, 0). fibonacci(1, 1). fibonacci(N, R) :-
```

$N > 1$,
 $N2$ is $N-2$,
 $N1$ is $N-1$,

fibonacci($N2$, $R2$),
fibonacci($N1$, $R1$),
 R is $R1+R2$.

%%
%% 6. /* Z é a soma dos números naturais X e Y */
soma(X,Y,Z) :- Z is $X + Y$.

%%
%% 7. fatorial(X,F) /* O fatorial de X é F */
fatorial(0, 1). fatorial(1, 1).
fatorial(X, F) :-
 $X > 1$, $X1$ is $X-$
 1, fatorial($X1$,
 $F1$), F is $X *$
 $F1$.

%%
%% 8. minimo($N1,N2,Min$) /* O mínimo de $N1$ e $N2$ é Min */
minimo($N1, N2, Min$) :-
 ($N1 < N2$, Min is $N1$);
 Min is $N2$.

%%
%% 9. mod(X,Y,Z) /* Z é o resto da divisão inteira de X por Y */
mod(X, Y, Z) :- Z is $X \bmod Y$.

%%
%% 10. sucessor(X,Y) /* X é sucessor do número Y */
sucessor(X, Y) :- X is $Y+1$.

%%
%% 11. entre(X,A,B) /* X está entre os números A e B */
entre(X, A, B) :- $X > A$, $X < B$.

12 - Questão

a)

$1+2==+(1,2)$ é verdadeira, pois a expressão “ $1+2$ ” e a expressão “ $+(1,2)$ ” são idênticas, então é o mesmo resultado.



b)

$1+2==2+1$ é falsa, pois a ordem dos elementos estão trocadas, ou seja, o conteúdo não é o mesmo.



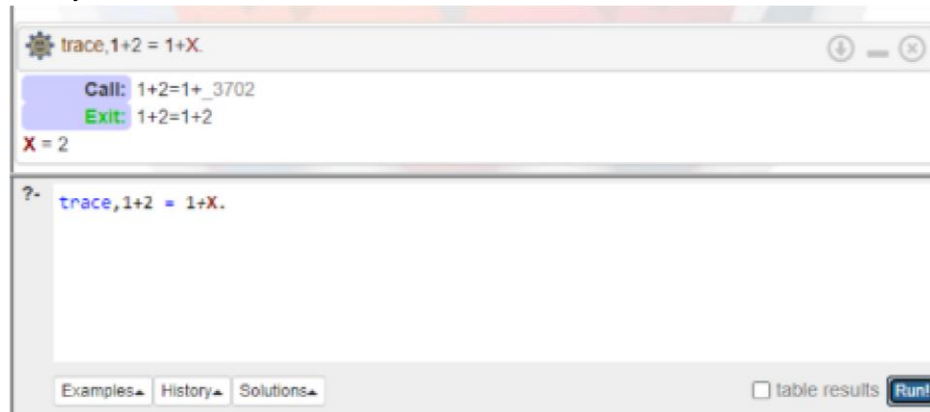
c)

$1+2==1+X$ é falsa, pois o conteúdo não é o mesmo das duas expressões pois X não está sendo inicializado com o valor 2.



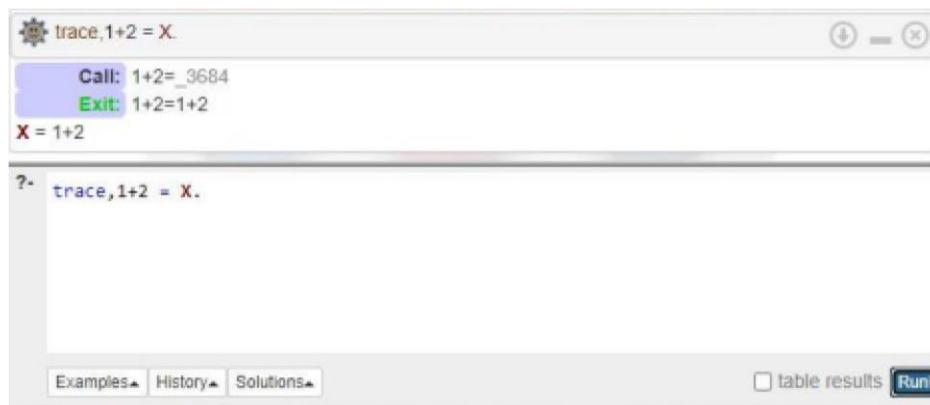
d)

As expressões são iguais, pois há o operador “=” o que significa que o X é instanciado com o valor 2, ou seja, resulta o mesmo valor.



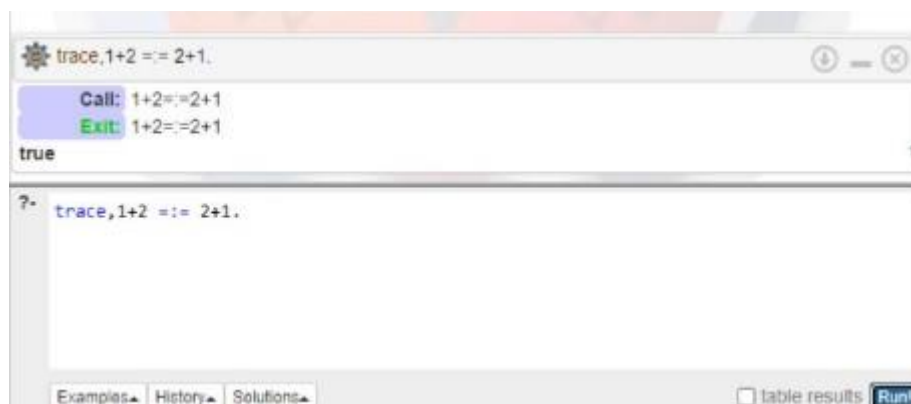
e)

$X=1+2$, pois X está sendo inicializado com a expressão 1+2 por causa do sinal “=”.



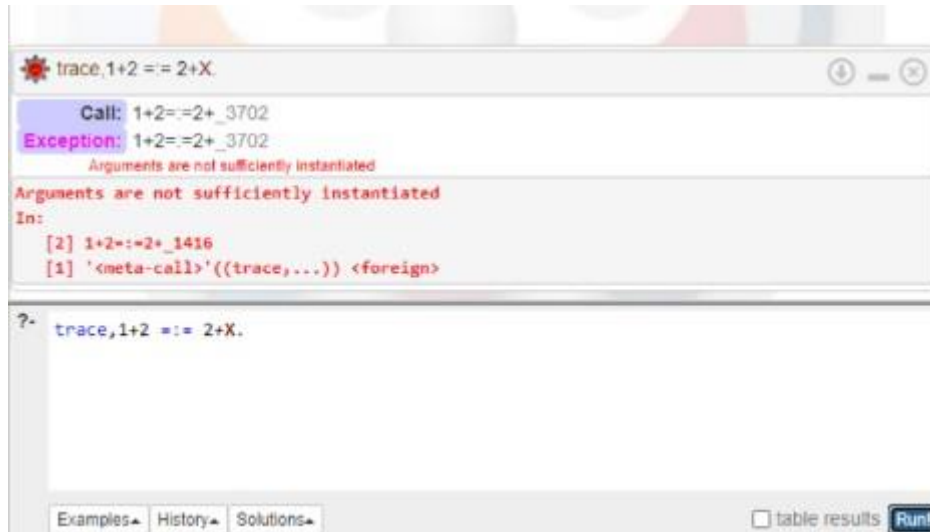
f)

$1+2:=2+1$ é verdadeira, pois as contas aritméticas são realizadas antes da análise do operador.



g)

$1+2 \text{ := } 2+X$ ocorre um erro, pois as contas aritméticas são feitas antes da análise do operador e o X não foi instanciado com um valor.



```
trace, 1+2 := 2+X.
```

Call: $1+2 \text{ := } 2+_3702$
Exception: $1+2 \text{ := } 2+_3702$
Arguments are not sufficiently instantiated
Arguments are not sufficiently instantiated
In:
[2] $1+2 \text{ := } 2+_1416$
[1] '<meta-call>'((trace,...)) <foreign>

?- trace, 1+2 := 2+X.

Examples History Solutions table results Run!

h)

$X=3$, pois o "is" inicializa o X com o resultado da conta aritmética.



```
trace, X is 1+2.
```

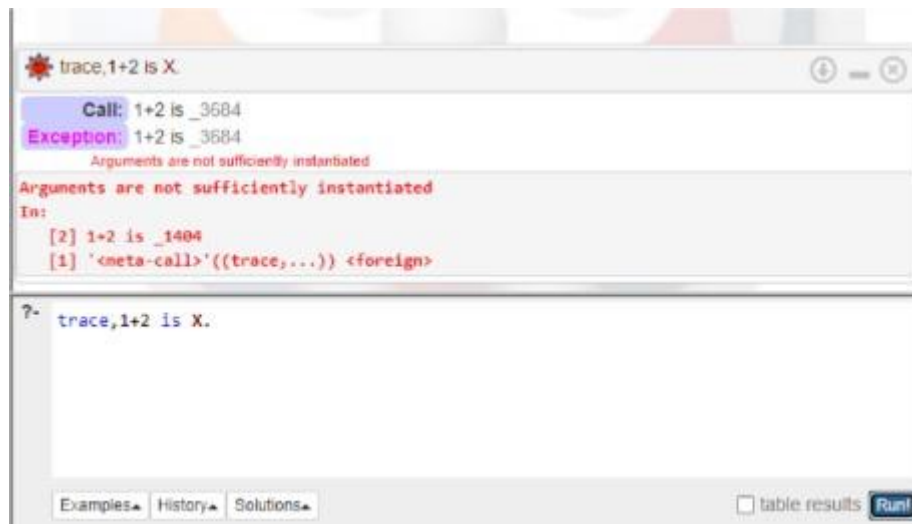
Call: $_3684 \text{ is } 1+2$
Exit: $3 \text{ is } 1+2$
 $X = 3$

?- trace, X is 1+2.

Examples History Solutions table results Run!

i)

$1+2 \text{ is } X$, ocorre um erro, pois está sendo instanciado uma incógnita em uma conta aritmética.



LINK DO VIDEO:

https://drive.google.com/file/d/1QBoachHj0gvCKt9wyMR7I1_UkRBgjmDQy/view?usp=sharing