

SolucionesExamenTipo-2020-cuaren...



Anónimo



Programación Declarativa: Lógica y Restricciones



3º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

ARF
15, 16 & 17.06.2023
Vitoria-Gasteiz

15.06
RANCID · EL DROGAS (BARRICADA - 40°)
16.06
THE PRETENDERS · INCUBUS
17.06
IGGY POP · LUCINDA WILLIAMS
AND MANY MORE...



AZKENAROCKFESTIVAL.COM



TARDARÍAS MENOS EN COMPRAR ESTE PORTÁTIL QUE TU ORDENADOR EN REINICIARSE. AHÍ LO DEJO.

**Lenovo**CONFIGURA EL EQUIPO A
MEDIDA EN [LENOVO.COM](https://lenovo.com)

Válido hasta el 30/06

PARTE I: PREGUNTAS TIPO TEST

Pregunta 1

Correcto

Calificado con
1.00 sobre 1.00[Editar pregunta](#)

Suponiendo que la estructura de datos con la que representaremos las matrices es una lista formada por las filas de la matriz (p.ej., $[[a,b],[c,d]]$). ¿Cuál sería el mejor caso base para el predicado recursivo $traspuesta/2$ ($traspuesta(X,Y)$), que se verifica si Y (matriz cuadrada) es la matriz traspuesta de X (matriz cuadrada)?

Seleccione una opción:

- ☐ a. $traspuesta([],[])$.
- ☐ b. $traspuesta([],[])$.
- ☒ c. $traspuesta([],[])$.
- ☐ d. $traspuesta([F],[F])$.

La respuesta correcta es: $traspuesta([],[])$.

Pregunta 2

Correcto

Calificado con
1.00 sobre 1.00[Editar pregunta](#)

¿Cuál sería el mejor caso base para el predicado recursivo $sumaLista/3$ ($sumaLista(N,L,S)$) que se verifica si S es la suma de los N primeros números naturales de la lista L ?

Seleccione una opción:

- ☐ a. $sumaLista(N,[],0)$.
- ☐ b. $sumaLista(0,[],0)$.
- ☐ c. $sumaLista([],0)$.
- ☒ d. $sumaLista(0,[],0)$.

La respuesta correcta es: $sumaLista(0,[],0)$.

Pregunta 3

Incorrecto

Calificado con
0.00 sobre 1.00[Editar pregunta](#)

¿Cuál sería el caso base para el predicado recursivo $mayor/2$ ($mayor(NP,X)$), que se verifica si NP (número en notación de Peano) es mayor que X (número en notación arábiga)?

Seleccione una opción:

- ☐ a. $mayor(A,B) :- A > B$.
- ☐ b. $mayor(s(),0)$.
- ☐ c. $mayor(A,0)$.
- ☒ d. $mayor(,0)$.

La respuesta correcta es: $mayor(s(),0)$.

Pregunta 4

Correcto

Calificado con
1.00 sobre 1.00[Editar pregunta](#)

Suponiendo el predicado puro $areaTriangulo/3$ ($areaTriangulo(B,H,A)$), que se verifica si B es la base de un triángulo, H es la altura de dicho triángulo, y A es el área del triángulo (calculada usando la base y la altura). ¿Cuál sería la consulta para saber cuánto vale el área de un triángulo de base 3 y altura 2?

Seleccione una opción:

- ☐ a. $?- areaTriangulo(3,2,A)$.
- ☐ b. $?- areaTriangulo(3,2,A)$.
- ☐ c. $?- areaTriangulo(s(s(s(0))),s(s(0)),area)$.
- ☒ d. $?- areaTriangulo(s(s(s(0))),s(s(0)),A)$.

La respuesta correcta es: $?- areaTriangulo(s(s(s(0))),s(s(0)),A)$.

Pregunta 5

Incorrecto

Calificado con
0.00 sobre 1.00[Editar pregunta](#)

¿Cuál sería el mejor caso base para el predicado recursivo $resto/3$ ($resto(X,Y,Z)$) que se verifica si Z es el resto de dividir X entre Y ?

Seleccione una opción:

- ☐ a. $resto(X,0,0)$.
- ☒ b. $resto(X,s(0),X)$.
- ☐ c. $resto(X,Y,X) :- menor(X,Y)$.
- ☐ d. $resto(X,Y,X) :- menor(X,Y)$.
- ☐ e. $resto(X,s(0),X)$.

La respuesta correcta es: $resto(X,Y,X) :- menor(X,Y)$.

RTX4000

legion
pro 5i

WUOLAH

Pregunta 6
Correcto
Calificado con
1.00 sobre 1.00

🚩

[Editar pregunta](#)

¿Qué consulta habría que realizar para obtener la lista de los elementos pares (LPares) de una lista L, en el mismo orden en el que aparecen en L?. Nota: en el caso de que no haya elementos pares, la consulta debe devolver una lista vacía.

Seleccione una opción:

- ☐ a. ?- setof(E,(member(E,L),par(E)),LPares).
- ☐ b. ?- bagof(E,(member(E,L),par(E)),LPares).
- ☒ c. ?- findall(E,(member(E,L),par(E)),LPares).

La respuesta correcta es: ?- findall(E,(member(E,L),par(E)),LPares).

Pregunta 7
Incorrecto
Calificado con
0.00 sobre 1.00

🚩

[Editar pregunta](#)

¿Qué función tiene el '!' en el siguiente predicado que obtiene el enésimo elemento de una lista?

```
nth(N, List, Elem) :-
    integer(N), !,
    N >= 1,
    nth_func(N, List, Elem).
nth(N, List, Elem) :-
    var(N), !,
    find_nth(List, Elem, 1, N).
```

Seleccione una opción:

- ☐ a. Corte verde
- ☒ b. Corte rojo
- ☐ c. No es un corte o esta sintaxis no está permitida
- ☐ d. Corte blanco

La respuesta correcta es: Corte blanco

Pregunta 8
Incorrecto
Calificado con
0.00 sobre 1.00

🚩

[Editar pregunta](#)

Suponiendo que el predicado recursivo listaFuntores/2 (listaFuntores(T,L)), que se verifica si T es un término Prolog y L es la lista de los funtores de aridad mayor o igual que 1 presentes en T, tiene dos casos base "listaFuntores(T,[])" ¿qué predicados meta-lógicos se deberían usar en dichos casos base?

Seleccione una opción:

- ☒ a. Ninguna de las otras respuestas
- ☐ b. compound(T) y var(T)
- ☐ c. ground(T)
- ☐ d. var(T) y atomic(T)
- ☐ e. atom(T) y ground(T)

La respuesta correcta es: var(T) y atomic(T)

Pregunta 9
Incorrecto
Calificado con
0.00 sobre 1.00

🚩

[Editar pregunta](#)

Se sabe que la consulta "?- X is 3/3, X = 1." falla, ¿cuál es el motivo?

Seleccione una opción:

- ☐ a. Porque en la llamada a 'is/2' X no está lo suficientemente instanciado
- ☐ b. Por un error de precisión de coma flotante en Prolog
- ☒ c. Porque se está unificando la X después, la query "1 is 3/3" no fallaría
- ☐ d. Porque el resultado de la división es un decimal y se unifica con un entero

La respuesta correcta es: Porque el resultado de la división es un decimal y se unifica con un entero

Pregunta 10
Incorrecto
Calificado con
0.00 sobre 1.00

🚩

[Editar pregunta](#)

Dado el siguiente programa, ¿por qué no termina la ejecución de la siguiente consulta: "?- list(_L), member(E,_L), nonvar(E).?"

```
list([]).
list([E|Es]) :- list(Es).
```

Seleccione una opción:

- ☐ a. Porque el caso base es incorrecto
- ☒ b. Porque no se está modificando ninguna variable en la cláusula recursiva y hay una recursión infinita
- ☐ c. La query termina pero es muy ineficiente
- ☐ d. Porque se generan infinitas soluciones, y todas ellas fallan

La respuesta correcta es: Porque se generan infinitas soluciones, y todas ellas fallan

PARTE II: EJERCICIOS CORTOS DE PROGRAMACIÓN

1. Programar el predicado lógico puro `even_list/1`, con cabecera `even_list(L)`, que se verifica si `L` es una lista de números de Peano pares de longitud par. Se prohíbe explícitamente el uso del predicado `length/2`, aunque se permite la definición de predicados auxiliares.

```
even(0).  
even(s(s(X))) :-  
    even(X).
```

```
even_list([]).  
even_list([X,Y|Xs]) :-  
    even(X),  
    even(Y),  
    even_list(Xs).
```

2. Programar el predicado ISO Prolog `unique/2`, con cabecera `unique(L,U)`, donde `U` es una lista ordenada de términos que aparecen una única vez en la lista `L`. El orden es el de aparición en la lista `L`.

```
unique(X,L) :-  
    findall(A,(select(A,X,R),\+member(A,R)),L).
```