**CANTIDAD DE HOJAS:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Texto  Descripción generada automáticamente con confianza baja | | | **UNIVERSIDAD NACIONAL DE HURLINGHAM** | | |
| **Inst. de Tecnología e Ingeniería**  LÓGICA Y PROGRAMACIÓN  Profesor: Mag. Ing. Pablo Pandolfo | | |
| Segundo Parcial junio 2022  ALUMNO: LU: FECHA: | | | | | |
| **NOTA: EL EXAMEN ESCRITO ES UN DOCUMENTO DE GRAN IMPORTANCIA PARA LA EVALUACIÓN DE LOS CONOCIMIENTOS ADQUIRIDOS, POR LO TANTO, SE SOLICITA LEER ATENTAMENTE LO SIGUIENTE:**   * Responda claramente cada punto, detallando con la mayor precisión posible lo solicitado. * Sea prolijo y ordenado en el desarrollo de los temas. * Sea cuidadoso con las faltas de ortografía y sus oraciones. * No desarrollar el examen en lápiz. * Aprobación del examen: Con nota mayor o igual a 4 (cuatro) * Condiciones de aprobación: 60% correcto. * Duración de examen: 2,5 horas. | | | | |  |
|  |  | **Ejercicio 1 [2]:** Desarróllese los hechos y regla/s que permita determinar si una persona vive en un estado vecino a otra persona no nordeste do Brasil.  Hombre con relleno sólidoHombre con relleno sólidoHombre con relleno sólidoHombre con relleno sólidoHombre con relleno sólidoHombre con relleno sólidoHombre con relleno sólidoHombre con relleno sólidoMapa  Descripción generada automáticamente    **Solución:**  vecinos(capital,sanagasta).  vecinos(chilecito,vinchina).  vecinos(sanagasta,capital)  vecinos(capital,sanagasta).  vecinos(chilecito,vinchina).  vecinos(sanagasta,capital)  vecinos(capital,sanagasta).  vecinos(chilecito,vinchina).  vecinos(sanagasta,capital)  vecinos(capital,sanagasta).  vecinos(chilecito,vinchina).  vecinos(sanagasta,capital)  vecinos(ma, pi).  vecinos(pi, ma). vecinos(pi, ce). vecinos(pi, pe). vecinos(pi, ba).  vecinos(ce, pi). vecinos(ce, rn). vecinos(ce, pb). vecinos(ce, pe).  vecinos(rn, ce). vecinos(rn, pb).  vecinos(pb, rn). vecinos(pb, ce). vecinos(pb, pe).  vecinos(pe, pb). vecinos(pe, ce). vecinos(pe, pi). vecinos(pe, ba). vecinos(pe, al).  vecinos(al, pe). vecinos(al, ba). vecinos(al, se).  vecinos(se, al). vecinos(se, ba).  vecinos(ba, pi). vecinos(ba, pe). vecinos(ba, se).  vive(ce, rojo). vive(pi, violeta). vive(rn, blanco). vive(pe, azul). vive(al, verde).  vive(ba, amarillo). vive(ba, gris). vive(ba, negro).  vive\_estado\_vecino(A, B) :- vive(A, X), vive(B, Y), vecinos(X, Y), A\==B.  vecinos(capital,sanagasta).  vecinos(chilecito,vinchina).  vecinos(sanagasta,capital)  **Ejercicio 2 [2]:** Los estados mencionados en el ejercicio anterior reciben determinada cantidad de visitantes al año. En base a este valor, determínese mediante una regla si son destinos turísticos atractivos o no, de acuerdo a si la cantidad de visitantes supera las 100000 personas.  turistas(al, 150000, 2021).  turistas(rn, 90000, 2022).  …  **Solución:**  turismo\_atractivo(X, Z) :- turistas(X, Y, Z), Y>100000.  **Ejercicio 3 [2]:** Genérese una lista a partir de ciertos valores, luego construir el árbol de resolución/ejecución. Ejemplo:  ?- intervalo(1, 5, L).  L = [1, 2, 3, 4, 5]  **Solución:**  Intervalo(X, X, [X]).  Intervalo(X, Y, [X | Xs]) :- X < Y, Z is X + 1, intervalo(Z, Y, Xs).  **Ejercicio 4 [2]:** Calcúlese detalladamente la precondición más débil para el siguiente algoritmo {pmd} aux = a; a = b; b = aux {a == y, b == x}  **Solución:**  wp(aux = a; a = b; b = aux, a == y, b == x)  wp(aux = a; a = b, wp(b = aux, a == y, b == x))  wp(aux = a; a = b, (a == y, b == x) {b/aux})  wp(aux = a; a = b, a == y, aux == x)  wp(aux = a, wp(a = b, a == y, aux == x))  wp(aux = a, (a == y, aux == x) {a/b))  wp(aux = a, b == y, aux == x)  (b == y, aux == x) {aux/a}  (b == y, a == x)  **Ejercicio 5 [2]:** Sean x e y variables enteras, calcúlese detalladamente la precondición más débil del siguiente fragmento de programa:  {pmd} si (x >= 0) entonces y = x sino y = -x {y == 4}  **Solución:**  wp(si (x >= 0) entonces y = x sino y = -x, y == 4)  (x >= 0 -> wp(y = x, y == 4)) ^ (¬ (x >= 0) -> wp(y = -x, y == 4))  (x >= 0 -> (y == 4) {y/x}) ^ (x < 0 -> (y == 4) {y/-x})  (x >= 0 -> x == 4) ^ (x < 0 -> -x == 4)  (|x| = 4) | |  | |