**CANTIDAD DE HOJAS:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | **UNIVERSIDAD DE BELGRANO** | | |
| **Facultad de Ingeniería y**  **Tecnología Informática**  LENGUAJES DE PROGRAMACIÓN (0050100023LENGP)  Profesor: Mag. Ing. Pablo Pandolfo | | |
| Examen Parcial octubre 2021  ALUMNO: LU: FECHA:  CARRERA: | | | | | |
| **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: 3 horas. | | | | |  | |
|  |  | Ejercicio 1 [2 puntos]: Resuélvase mediante programación lógica el siguiente problema: Un agente comercial debe visitar N ciudades dadas minimizando el numero total de km recorridos. Puede elegir el orden en el que visita las ciudades, en particular donde comenzar y donde terminar. Para la planificación el agente comercial dispone de una lista de ciudades a visitar, expresadas como ciudades([mdq, igz, bar]) y una tabla precalculada con las distancias mínimas entre cada para de ciudades, representadas mediante: dist(igz, bar, 3200). dist(bar, igz, 3200). dist(igz, mdq, 2000). Dist(mdq, igz, 2000). Denominamos recorrido a una lista Prolog que contiene las N ciudades en el orden en que se visitan. Escríbase un predicado dist\_total(R, D) que dado un recorrido R, instancie la variable D a la distancia total de dicho recorrido.  **dist\_total([], 0).**  **dist\_total([X,Y | L] , T) :- dist(X, Y, D), dist\_total([Y | L], U), T is D+U.**  Ejercicio 2 [2 puntos]: Diséñese las reglas de la GIC en formato BNF para describir la sintaxis de la siguiente sentencia: Un cierto lenguaje de programación utiliza una función TAKE. Esta función toma como entrada un operador de suma, resta, multiplicación o división entera, una constante entera y una lista, devolviendo el valor que resulta de aplicar el operador a los primeros “n” elementos de la lista. El valor de n quedará establecido en la componente cte. La lista estará conformada por constantes separadas por coma (,) y delimitadas por corchetes. La lista puede ser vacía. Formato de la función: TAKE (Operador; constante; [lista de constantes])  Ejemplos:  TAKE (\* ; CTE ; [ CTE , CTE, CTE] )  TAKE (- ; CTE ; [ CTE ] )  TAKE (/ ; CTE ; [ ] )  Considerar las constantes como símbolos terminales (CTE) (ej. corchetes, paréntesis, operadores aritméticos, punto y coma).  **<funciontake> ::= TAKE ( <operador> ; CTE ; [<listacte>]) | TAKE (<operador> ; CTE ; [])**  **<listacte> ::= CTE | <listacte> , CTE**  **<operador> ::= + | - | \* | /**  Ejercicio 3 [2 puntos]: Diséñese el Diagrama de Sintaxis, del lenguaje Lambda Calculus.  **<expr> ::= <var> | λ <var> . <expr> | <expr> <expr> | ( <expr> )**  **<var> ::= x | y | z | f | n | <var> ´**  Imagen de la pantalla de un celular con letras  Descripción generada automáticamente con confianza media  Ejercicio 4 [2 punto]: Obténgase el lambda calculus de la expresión **and T T** sabiendo que and es λa. λb. a b a “y” T es λx. λy. x  **λa. λb. a b a T T**  **λb. T b T T**  **λb. λx. λy. x b T T**  **λb.** **λy. b T T**  **λy. T T**  **λy. λx. λy. x T**  **λx. λy. x**  Ejercicio 5 [1 punto]: Calcúlese {P}  {P} IF X = 0 THEN X ::= Y + 1 ELSE Z ::= Y + 1 {(X = Y + 1) | (Z = X + 1)}  **P = {X = Y}**  Ejercicio 6 [1 punto]: Justifíquese: Python es un lenguaje de programación multiparadigma.  **Soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional.** | |  |