**INSTITUTO TECNOLÓGICO DE COSTA RICA I SEMESTRE 2018**

**INGENIERÍA EN COMPUTACIÓN**

**IC-4700 LENGUAJES DE PROGRAMACIÓN**

**DOCUMENTACIÓN TAREA PROGRAMADA 1 - HASKEL**

1. Introducción

Presentar el problema. Puede “reciclar” partes del enunciado de la tarea programada.

1. Estructuras de datos usadas

Describir las estructuras de datos usadas para implementar el ambiente: árbol, parámetros, ambiente. Indicar si dichas estructuras de datos están ordenadas por algún criterio o no y cuál es ese criterio.

1. Instrucciones para ejecutar el programa

Presentar la sintaxis concreta usada en los comandos. Además debe explicar cómo debe ser arrancado el programa y cómo se deben entrar los diferentes comandos.

1. Corridas de ejemplo

Mostrar el resultado de las corridas de prueba que se incluyen al final de este documento. En caso de implementación parcial de la tarea se pueden mostrar la ejecución de aquellas partes del programa que funcionan.

1. Comentarios finales (estado del programa)

Indicar el estado final en que quedó el programa, problemas encontrados y limitaciones adicionales.

8. Directorio con documentación

La tarea debe ser entregada presentando un archivo comprimido (ZIP) que incluya la siguiente estructura:

TP1-*Apellidos-Nombre* (directorio)

*Archivo\_de\_documentacion*

*Programas*

*Archivos\_adicionales*

9. Entrega

Enviar la tarea y su documentacion a la direccion [josee.arayamonge@gmail.com](mailto:josee.arayamonge@gmail.com). La fecha límite es el jueves 26 de abril a las 1:00pm.

**Corridas de prueba**

**>> ie x1 = y + z**

**{ x1, [y, z], "y + z", [y, z], "y + z" }**

**>> ie x1 = 1 + n**  *// rechazar, x1 ya está definida*

**>> ie x = x1 \* w**

**{ x, [x1, w], "x1 \* w", [y, z, w], "(y + z) \* w "}**

**>> mv x**

**{ x, [x1, w], "x1 \* w", [y, z, w], "(y + z) \* w "}**

**>> cv x 2 3 4)**  *// y=2 z=3 w=4*

**20**

**>> ie u = w – t**

**{ u, [w, t], "w - t", [w, t], "w - t" }**

**>> mp**

**y z w t** *// el orden puede variar*

**>> et 2 3 4 1**  *// y=2 z=3 w=4 t=1*

**[ (x1,5), (x,20), (u,3) ]**

**>> ie z = a \* b**

**{ z, [a, b], "a \* b", [a, b], "a \* b "}**

**>> ma**

**{ x1, [y, z], "y + z", [y, a, b], "y + (a \* b)" }**

**{ x, [x1, w], "x1 \* w", [y, a, b, w], "(y + (a \* b)) \* w "}**

**{ u, [w, t], "w - t", [w, t], "w - t" }**

**{ z, [a, b], "a \* b", [a, b], "a \* b "}**

**>> cv x 2 3 4 5**  *// y=2 a=3 b=4 w=5 🡺 (2 + (3 \* 4)) \*5 =70*

**70**

**>> ie w = 2 \* u** *// rechazar, se forma ciclo: u depende de w, w depende de u*

**>> ie w = 2 \* z** *// z modifica a t, t modifica a u*

**{ w, [z], "2 \* z", [a, b], "2 \* (a \* b)" }**

**>> ma**

**{ x1, [y, z], "y + z", [y, a, b], "y + (a\*b)" }**

**{ x, [x1, w], "x1 \* w", [y, a, b], "(y + (a\*b)) \* (2 \* (a\*b)) "}**

**{ u, [w, t], "w - t", [w, a, b], "(2 \* (a\*b)) - t" }**

**{ z, [a, b], "a \* b", [a, b], "a\*b"}**

**{ w, [z], "2 \* z", [a, b], "2 \* (a\*b)" }**

**>> mp**

**y a b t** *// el orden puede variar*

**>> et 2 3 4 5** *// y=2 a=3 b=4 t=5*

**[ (x1,14), (x,336), (u,19), (z,12), (w,24) ]**

**>> u = r \* s** *// rechazar, u ya está definida*