

Pregunta 1-a:

- **Tipos de alcances y asociaciones:** Mercury es un lenguaje de programación funcional y lógico que utiliza un sistema de alcance estático y asociación léxica. Una ventaja de este sistema de alcance y asociación es que permite una gestión más clara y predecible de las variables en el código. Sin embargo, una desventaja de este sistema es que puede ser menos flexible que otros sistemas de alcance y asociación, como el alcance dinámico y la asociación dinámica. En general, el sistema de alcance y asociación de Mercury es adecuado para los usuarios objetivos del lenguaje, que son programadores que buscan un lenguaje de programación funcional y lógico con un alto nivel de seguridad y predictibilidad en el manejo de variables. Los diseñadores del lenguaje han tomado una decisión consciente de priorizar la seguridad y la predictibilidad sobre la flexibilidad en este aspecto del lenguaje, y esto se refleja en la elección del sistema de alcance y asociación estático y léxico.
- **Módulos:** Mercury proporciona una amplia gama de módulos estándar que contienen implementaciones comunes de estructuras de datos y algoritmos. Estos módulos están destinados a ser utilizados en una variedad de aplicaciones y cubren áreas como listas, conjuntos, mapas, entrada/salida, aritmética, etc. Los programadores también pueden crear sus propios módulos para organizar su código de manera modular. Estos módulos de usuario pueden contener predicados, funciones, tipos y otros elementos del lenguaje.
- **Importación y exportación de nombres:** Los nombres pueden ser importados selectivamente, calificadamente o por defecto, y pueden ser exportados explícita o implícitamente en los módulos de usuario, lo que proporciona flexibilidad en la organización y estructuración del código. Mercury admite la sobrecarga de predicados y funciones. La sobrecarga permite definir múltiples predicados o funciones con el mismo nombre pero con diferentes aridades o tipos de argumentos. El compilador de Mercury seleccionará automáticamente la versión correcta del predicado o función basándose en los tipos de los argumentos utilizados en la llamada. Mercury también admite el polimorfismo paramétrico, que permite escribir código que es genérico en cuanto al tipo de datos que manipula. Esto facilita la reutilización del código para diferentes tipos de datos.
- **Herramientas de Mercury:**
 - **Compiladores:** Mercury Compiler (mmc) es el compilador principal de Mercury. Puede compilar código Mercury en código máquina o en código intermedio para ser interpretado por el sistema de tiempo de ejecución de Mercury.

Pregunta 2:

X = 1

$$Y = 2$$

Z = 6

a) Alcance estático y asociación profunda

Al culminar la ejecución de $s(c * a, R)$:

[illegible]

Al ejecutar $r(c+a)$ en P.Q:

[illegible]

Al finalizar la ejecución de $t(c + b)$ dentro de $Q(c^*b, t)$:

[illegible]

Al salir de Q y de el condicional if-else:

[illegible]

Y se imprime:

27 1215 31

9 1215 13

9 1215 9

- **c) Alcance estático y asociación superficial:**

Luego de la ejecución de $s(c * a, R)()$:

[illegible]

Luego de la ejecución de $r(c+a)$:

[illegible]

Luego de la ejecución de $t(c+b)$:

[illegible]

La primera salida seria:

[illegible]

Se imprime: