#### 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):

					23				
					22				
					21				
					20				
					19				
					18				
					17	b	19	G	P1.R
					16	а	9		PI.R
					15	b	993	G	
					14	r	R	9	Q
					13	а	992		
Constante	Valor	Claus	suras		12	а	32		
Х	1	Р	b	Global	11	С	31		
Y	2	r	С	Global	10	Q	proc		
Z	6	Q	b	Global	9	R	proc		P1
		Q	С	Global	8	t	R	3	
		R	а	Global	7	S	Q	G	
Gle	obal	K	С	Global	6	а	27		
Р	proc	P0.R	b	Global	5	С	13		
Q	proc	FU.R	С	Global	4	Q	proc		
R	proc	P1.R	b	Global	3	R	proc		P0
С	9	r I.R	С	Global	2	t	R	G	FU
b	19				1	S	Q	G	
а	9				0	а	9		

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

					23				
					22				
					21				
					20				
					19				
					18				
					17	b	653		D0 D
					16	а	643		P0.R
					15	С	616		
					14	r	R	3	P1.Q
					13	b	589		
Constante	Valor	Claus	suras		12	а	32		
Х	1	Р	b	Global	11	С	31		
Υ	2	P	С	Global	10	Q	proc		
z	6	Q	b	Global	9	R	proc		P1
		Q	С	Global	8	t	R	3	
		R	а	Global	7	S	Q	G	
Glo	bal	K	С	Global	6	а	27		
Р	proc	P0.R	b	Global	5	С	13		
Q	proc	PU.K	С	Global	4	Q	proc		
R	proc		С	Global	3	R	proc		P0
С	9	P1.Q	а	6	2	t	R	G	P0
b	653		t	8	1	s	Q	G	
а	9				0	а	9		

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

					23				
					22				
					21				
					20				
					19				
					18				
					17	b	1215		D
					16	а	1205		P0.R
					15	С	616		
					14	r	R	3	P1.Q
					13	b	589		
Constante	Valor	Clau	suras		12	а	32		
Х	1	P	b	Global	11	С	31		
Υ	2	P	С	Global	10	Q	proc		
z	6	0	b	Global	9	R	proc		P1
		Q	С	Global	8	t	R	3	
		R	а	Global	7	s	Q	G	
Gle	obal	K	С	Global	6	а	27		
Р	proc	P0.R	b	Global	5	С	13		
Q	proc	ru.K	С	Global	4	Q	proc		
R	proc		С	Global	3	R	proc		P0
С	9	P1.Q	а	6	2	t	R	G	FU
b	1215		t	8	1	s	Q	G	
а	9				0	а	9		

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

					_					
						23				
					:	22				
						21				
						20				
						19				ĺ
						18				
						17				
						16				ĺ
						15				
						14				
						13				
Constante	Valor					12				
X	1					11	С	31		
Y	2					10	Q	proc		
Z	6					9	R	proc		P1
						8	t	R	3	
		Cla	ausuras			7	s	Q	G	
Gl	obal	Р	b	Global		6	а	27		
Р	proc		С	Global		5	С	13		
Q	proc	Q	b	Global		4	Q	proc		
R	proc	ď	С	Global		3	R	proc		P0
С	9	R	а	Global		2	t	R	G	
b	1215	K	С	Global		1	s	Q	G	
а	9					0	а	9		

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) ():

					23				
					22				
					21				
					20				
					19				
					18				
					17	b	19	G	P.R0
					16	а	9		F.KU
					15	b	993	G	
					14	r	R	9	QG0
					13	а	992		
Constante	Valor				12	а	32		
X	1				11	С	31		
Υ	2				10	Q	proc		
Z	6				9	R	proc		P1
					8	t	R	3	
					7	s	Q	G	
Glo	bal	Clau	suras		6	а	27		
Р	proc	Р	а	Global	5	С	13		
Q	proc	Q	С	Global	4	Q	proc		
R	proc	3	b	Global	3	R	proc		P0
С	9	R	b	Global	2	t	R	G	P
b	19	IX.	С	Global	1	s	Q	G	
а	9				0	а	9		

#### Luego de la ejecución de r(c+a):

					23				
					22				
					21				
					20				
					19				
					18				
					17	b	616		P0.R
					16	а	607		FU.K
					15	С	598	G	
					14	r	P0.R	3	P1.Q
					13	b	589		
Constante	Valor				12	а	32		
Х	1				11	С	31		
Υ	2				10	Q	proc		
Z	6				9	R	proc		P1
					8	t	R	3	
					7	s	Q	G	
Glo	bal	Clau	suras		6	а	27		
Р	proc	Р	а	Global	5	С	13		
Q	proc	P1.Q	а	0	4	Q	proc		
R	proc	F1.Q	С	Global	3	R	proc		P0
С	9	P0.R	b	Global	2	t	R	G	Ρ0
b	616	FU.K	С	Global	1	s	Q	G	
а	9				0	а	9		

Luego de la ejecución de t(c+b):

					$\overline{}$			_	
					23				
					22				
					21				
					20				
					19				
					18				
					17	b	1224		DO D
					16	а	1214		P0.R
					15	С	598	G	
					14	r	P0.R	3	P1.Q
					13	b	589		
Constante	Valor				12	а	32		
Х	1				11	С	31		
Y	2				10	Q	proc		
Z	6				9	R	proc		P1
					8	t	R	3	
					7	s	Q	G	
Glol	oal	Clau	suras		6	а	27		
Р	proc	Р	а	Global	5	С	13		
Q	proc		а	0	4	Q	proc		
R	proc	P1.Q	С	Global	3	R	proc		D0
С	9				2	t	R	G	P0
b	1224	Do D	b	Global	1	s	Q	G	
а	9	P0.R	С	Global	0	а	9		

#### La primera salida seria:

#### 27 1224 31

# Se desempila P1

					23				
					22				
					21				
					20				
					19				
					18				
					17				
					16				]
					15				
					14				1
					13				1
Constante	Valor				12				
Х	1				11				
Υ	2				10				
Z	6				9				
					8				
					7				
Glo	bal	Claus	suras		6				
Р	proc	Р	а	Global	5	С	13		
Q	proc		а	0	4	Q	proc		
R	proc	P1.Q	С	Global	3	R	proc		D0
С	9				2	t	R	G	P0
b	1224	DO D	b	Global	1	s	Q	G	
а	9	P0.R	С	Global	0	а	9		

## Se imprime:

9 1224 13

9 1224 9