

La conjetura de Ulam

En lenguaje R

Integrantes

Castillo Reyes Eduardo Armando
Maravilla Pérez Vianey
Vázquez Portuguese José Antonio

Docente

M. en C. Cristal Karina Galindo Durán

4 de septiembre del 2021

Tabla de contenido

Problema	1
Consideraciones	1
Diagrama de flujo	2
Pseudocódigo	3
Pruebas de escritorio	4
Conclusión	5

Problema

La conjetura de Collatz, conocida también como conjetura $3n+1$ o conjetura de Ulam (entre otros nombres), fue enunciada por el matemático Lothar Collatz en 1937, y a la fecha no se ha resuelto.

Enunciado

Sea la siguiente operación, aplicable a cualquier número entero positivo:

- Si el número es par, se divide entre 2.
- Si el número es impar, se multiplica por 3 y se suma 1.

Formalmente, esto equivale a una función: $f: N \rightarrow N$

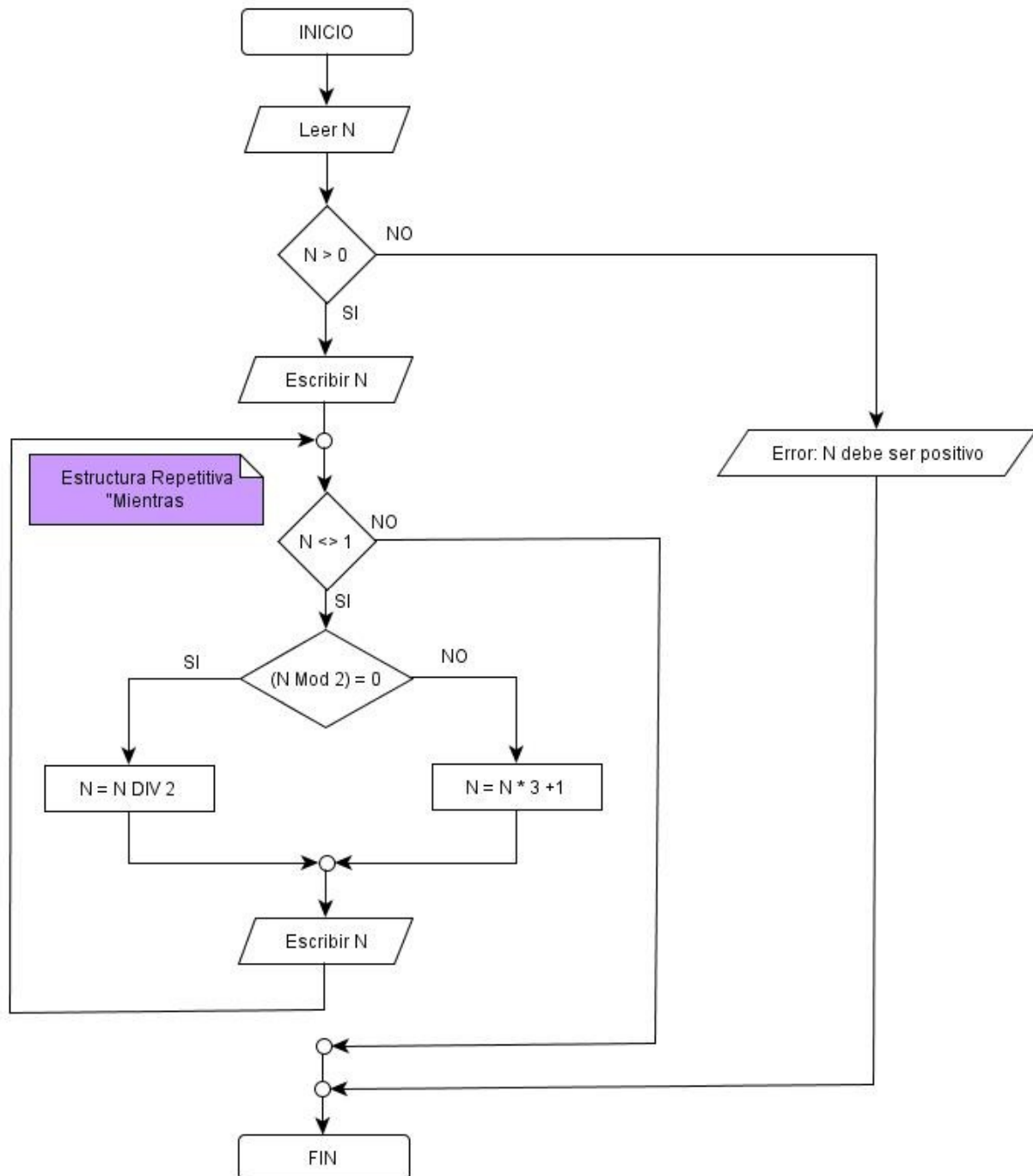
$$f(n) = \begin{cases} \frac{n}{2}, & \text{si } n \text{ es par} \\ 3n + 1, & \text{si } n \text{ es impar} \end{cases}$$

Consideraciones

Para la validación del correcto funcionamiento del programa se utilizaron las siguientes restricciones en el programa:

- Solo entrada/lectura de números Naturales
- Solo números menores a 2^{32} (tamaño de Integer)

Diagrama de flujo



Pseudocódigo

Escribir "Favor ingresar el número: "

Leer numero

valor <- número

Mientras(valor > 1)

 Si (Valor 2 = 0) Entonces

 valor <- trunc(valor / 2)

 Escribir valor

 Sino

 valor = (valor * 3) + 1

 Escribir valor

 FinSi

FinMientras

Escribir "El numero: ", numero, "Tiene como conjetura de ulam consecutivamente: ", valor

FinProceso

Pruebas de escritorio

Prueba de escritorio 1. Número = 7									
Descripción de sentencia	Sentencia	n	isValid	inicial	numeros[]	contador	show	numeros	
Lectura de numero	n <- suppressWarnings(as.integer(n))	7							
Validación de numero y mayor a 1	isValid <- !is.na(n) && n > 1	true							
Declaración de inicial con valor n	inicial <- n			7					
Declaración de vector con limite de 1000	numeros <- numeric(limit=1000)				[]				
Declaración de contador con valor 0	contador <- 0					0			
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit								
Aumento de contador	contador <- contador + 1					1			
Asignación de valor en el vector	numeros[contador] <- n				[1]=7				
Llamada a la funcion ulam	n <- ulamRegla(n)								
Condición de residuo par igual a 0	m %% 2 == 0								
Retorna el valor a la mitad	return(m/2)								
Retorna el valor al triple más uno	return(3*m + 1)	22							
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit								
Aumento de contador	contador <- contador + 1					2			
Asignación de valor en el vector	numeros[contador] <- n				[2] = 22				
Llamada a la funcion ulam	n <- ulamRegla(n)								
Condición de residuo par igual a 0	m %% 2 == 0								
Retorna el valor a la mitad	return(m/2)	11							
Retorna el valor al triple más uno	return(3*m + 1)								
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit								
Aumento de contador	contador <- contador + 1					3			
Asignación de valor en el vector	numeros[contador] <- n				[3] = 11				
Llamada a la funcion ulam	n <- ulamRegla(n)								
Condición de residuo par igual a 0	m %% 2 == 0								
Retorna el valor a la mitad	return(m/2)								
Retorna el valor al triple más uno	return(3*m + 1)	34							
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit								
Aumento de contador	contador <- contador + 1					4			
Asignación de valor en el vector	numeros[contador] <- n				[4] = 34				
Llamada a la funcion ulam	n <- ulamRegla(n)								
Condición de residuo par igual a 0	m %% 2 == 0								
Retorna el valor a la mitad	return(m/2)	17							
Retorna el valor al triple más uno	return(3*m + 1)								
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit								
Aumento de contador	contador <- contador + 1					5			
Asignación de valor en el vector	numeros[contador] <- n				[5] = 17				
Llamada a la funcion ulam	n <- ulamRegla(n)								
Condición de residuo par igual a 0	m %% 2 == 0								
Retorna el valor a la mitad	return(m/2)								
Retorna el valor al triple más uno	return(3*m + 1)	52							
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit								
Aumento de contador	contador <- contador + 1					6			
Asignación de valor en el vector	numeros[contador] <- n				[6] = 52				
Llamada a la funcion ulam	n <- ulamRegla(n)								
Condición de residuo par igual a 0	m %% 2 == 0								
Retorna el valor a la mitad	return(m/2)	26							
Retorna el valor al triple más uno	return(3*m + 1)								
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit								
Aumento de contador	contador <- contador + 1					7			
Asignación de valor en el vector	numeros[contador] <- n				[7] = 26				
Llamada a la funcion ulam	n <- ulamRegla(n)								
Condición de residuo par igual a 0	m %% 2 == 0								
Retorna el valor a la mitad	return(m/2)	13							
Retorna el valor al triple más uno	return(3*m + 1)								
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit								
Aumento de contador	contador <- contador + 1					8			
Asignación de valor en el vector	numeros[contador] <- n				[8] = 13				
Llamada a la funcion ulam	n <- ulamRegla(n)								
Condición de residuo par igual a 0	m %% 2 == 0								
Retorna el valor a la mitad	return(m/2)								
Retorna el valor al triple más uno	return(3*m + 1)	40							
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit								
Aumento de contador	contador <- contador + 1					9			
Asignación de valor en el vector	numeros[contador] <- n				[9] = 40				
Llamada a la funcion ulam	n <- ulamRegla(n)								
Condición de residuo par igual a 0	m %% 2 == 0								
Retorna el valor a la mitad	return(m/2)	20							
Retorna el valor al triple más uno	return(3*m + 1)								
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit								

Aumento de contador	contador <- contador + 1			10	
Asignación de valor en el vector	numeros[contador] <- n		[10] = 20		
Llamada a la funcion ulam	n <- ulamRegla(n)				
Condicion de residuo par igual a 0	m %% 2 == 0				
Retorna el valor a la mitad	return(m/2)	10			
Retorna el valor al triple más uno	return(3*m + 1)				
Condicion de paro con n mayor a 1 y contador menor al límite	n > 1 & contador < limit				
Aumento de contador	contador <- contador + 1			11	
Asignación de valor en el vector	numeros[contador] <- n		[11] = 10		
Llamada a la funcion ulam	n <- ulamRegla(n)				
Condicion de residuo par igual a 0	m %% 2 == 0				
Retorna el valor a la mitad	return(m/2)	5			
Retorna el valor al triple más uno	return(3*m + 1)				
Condicion de paro con n mayor a 1 y contador menor al límite	n > 1 & contador < limit				
Aumento de contador	contador <- contador + 1			12	
Asignación de valor en el vector	numeros[contador] <- n		[12] = 5		
Llamada a la funcion ulam	n <- ulamRegla(n)				
Condicion de residuo par igual a 0	m %% 2 == 0				
Retorna el valor a la mitad	return(m/2)				
Retorna el valor al triple más uno	return(3*m + 1)	16			
Condicion de paro con n mayor a 1 y contador menor al límite	n > 1 & contador < limit				
Aumento de contador	contador <- contador + 1			13	
Asignación de valor en el vector	numeros[contador] <- n		[13] = 16		
Llamada a la funcion ulam	n <- ulamRegla(n)				
Condicion de residuo par igual a 0	m %% 2 == 0				
Retorna el valor a la mitad	return(m/2)	8			
Retorna el valor al triple más uno	return(3*m + 1)				
Condicion de paro con n mayor a 1 y contador menor al límite	n > 1 & contador < limit				
Aumento de contador	contador <- contador + 1			14	
Asignación de valor en el vector	numeros[contador] <- n		[14] = 8		
Llamada a la funcion ulam	n <- ulamRegla(n)				
Condicion de residuo par igual a 0	m %% 2 == 0				
Retorna el valor a la mitad	return(m/2)	4			
Retorna el valor al triple más uno	return(3*m + 1)				
Condicion de paro con n mayor a 1 y contador menor al límite	n > 1 & contador < limit				
Aumento de contador	contador <- contador + 1			15	
Asignación de valor en el vector	numeros[contador] <- n		[15] = 4		
Llamada a la funcion ulam	n <- ulamRegla(n)				
Condicion de residuo par igual a 0	m %% 2 == 0				
Retorna el valor a la mitad	return(m/2)	2			
Retorna el valor al triple más uno	return(3*m + 1)				
Condicion de paro con n mayor a 1 y contador menor al límite	n > 1 & contador < limit				
Aumento de contador	contador <- contador + 1			16	
Asignación de valor en el vector	numeros[contador] <- n		[16]=2		
Llamada a la funcion ulam	n <- ulamRegla(n)				
Condicion de residuo par igual a 0	m %% 2 == 0				
Retorna el valor a la mitad	return(m/2)	1			
Retorna el valor al triple más uno	return(3*m + 1)				
Encuentra el mínimo entre contador y limit	howMany <- min(contador, limit)				
Cuadro de texto para visualizar grafica	show <- readline("Quieres visualizarla (y/n)? ")			y	
Condición de valor en show igual a 'y'	show == "y"				
Impresión de los numeros	print(numeros[1:howMany])				7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2

Prueba de escritorio 1. Número = 30							
Descripción de sentencia	Sentencia	n	isValid	inicial	numeros[]	contador	show numeros
Lectura de numero	n <- suppressWarnings(as.integer(n))	7					
Validación de numero y mayor a 1	isValid <- !is.na(n) && n > 1		true				
Declaración de inicial con valor n	inicial <- n			7			
Declaración de vector con limite de 1000	numeros <- numeric(limit=1000)				[]		
Declaración de contador con valor 0	contador <- 0					0	
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1					1	
Asignación de valor en el vector	numeros[contador] <- n				[1]=30		
Llamada a la función ulam	n <- ulamRegla(n)						
Condición de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	15					
Retorna el valor al triple más uno	return(3*m + 1)						
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1					2	
Asignación de valor en el vector	numeros[contador] <- n				[2] = 15		
Llamada a la función ulam	n <- ulamRegla(n)						
Condición de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)						
Retorna el valor al triple más uno	return(3*m + 1)	46					
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1					3	
Asignación de valor en el vector	numeros[contador] <- n				[3] = 46		
Llamada a la función ulam	n <- ulamRegla(n)						
Condición de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	23					
Retorna el valor al triple más uno	return(3*m + 1)						
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1					4	
Asignación de valor en el vector	numeros[contador] <- n				[4] = 23		
Llamada a la función ulam	n <- ulamRegla(n)						
Condición de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)						
Retorna el valor al triple más uno	return(3*m + 1)	70					
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1					5	
Asignación de valor en el vector	numeros[contador] <- n				[5] = 70		
Llamada a la función ulam	n <- ulamRegla(n)						
Condición de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	35					
Retorna el valor al triple más uno	return(3*m + 1)						
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1					6	
Asignación de valor en el vector	numeros[contador] <- n				[6] = 35		
Llamada a la función ulam	n <- ulamRegla(n)						
Condición de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)						
Retorna el valor al triple más uno	return(3*m + 1)	106					
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1					7	
Asignación de valor en el vector	numeros[contador] <- n				[7] = 106		
Llamada a la función ulam	n <- ulamRegla(n)						
Condición de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	53					

Retorna el valor al triple más uno	return(3*m + 1)			
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit			
Aumento de contador	contador <- contador + 1			
Asignación de valor en el vector	numeros[contador] <- n	[8] = 53	8	
Llamada a la funcion ulam	n <- ulamRegla(n)			
Condicion de residuo par igual a 0	m %% 2 == 0			
Retorna el valor a la mitad	return(m/2)			
Retorna el valor al triple más uno	return(3*m + 1)		160	
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit			
Aumento de contador	contador <- contador + 1			
Asignación de valor en el vector	numeros[contador] <- n	[9] = 160	9	
Llamada a la funcion ulam	n <- ulamRegla(n)			
Condicion de residuo par igual a 0	m %% 2 == 0			
Retorna el valor a la mitad	return(m/2)		80	
Retorna el valor al triple más uno	return(3*m + 1)			
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit			
Aumento de contador	contador <- contador + 1			
Asignación de valor en el vector	numeros[contador] <- n	[10] = 80	10	
Llamada a la funcion ulam	n <- ulamRegla(n)			
Condicion de residuo par igual a 0	m %% 2 == 0			
Retorna el valor a la mitad	return(m/2)		40	
Retorna el valor al triple más uno	return(3*m + 1)			
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit			
Aumento de contador	contador <- contador + 1			
Asignación de valor en el vector	numeros[contador] <- n	[11] = 40	11	
Llamada a la funcion ulam	n <- ulamRegla(n)			
Condicion de residuo par igual a 0	m %% 2 == 0			
Retorna el valor a la mitad	return(m/2)		20	
Retorna el valor al triple más uno	return(3*m + 1)			
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit			
Aumento de contador	contador <- contador + 1			
Asignación de valor en el vector	numeros[contador] <- n	[12] = 20	12	
Llamada a la funcion ulam	n <- ulamRegla(n)			
Condicion de residuo par igual a 0	m %% 2 == 0			
Retorna el valor a la mitad	return(m/2)		10	
Retorna el valor al triple más uno	return(3*m + 1)			
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit			
Aumento de contador	contador <- contador + 1			
Asignación de valor en el vector	numeros[contador] <- n	[13] = 10	13	
Llamada a la funcion ulam	n <- ulamRegla(n)			
Condicion de residuo par igual a 0	m %% 2 == 0			
Retorna el valor a la mitad	return(m/2)		5	
Retorna el valor al triple más uno	return(3*m + 1)			
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit			
Aumento de contador	contador <- contador + 1			
Asignación de valor en el vector	numeros[contador] <- n	[14] = 5	14	
Llamada a la funcion ulam	n <- ulamRegla(n)			
Condicion de residuo par igual a 0	m %% 2 == 0			
Retorna el valor a la mitad	return(m/2)		16	
Retorna el valor al triple más uno	return(3*m + 1)			
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit			
Aumento de contador	contador <- contador + 1			
Asignación de valor en el vector	numeros[contador] <- n	[15] = 16	15	
Llamada a la funcion ulam	n <- ulamRegla(n)			
Condicion de residuo par igual a 0	m %% 2 == 0			
Retorna el valor a la mitad	return(m/2)		8	
Retorna el valor al triple más uno	return(3*m + 1)			
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit			
Aumento de contador	contador <- contador + 1			
Asignación de valor en el vector	numeros[contador] <- n	[16]=8	16	
Llamada a la funcion ulam	n <- ulamRegla(n)			
Condicion de residuo par igual a 0	m %% 2 == 0			

Retorna el valor a la mitad	return(m/2)	4	
Retorna el valor al triple más uno	return(3*m + 1)		
Condición de paro con n mayor a 1 y contador menor al límite	n > 1 & contador < limit		
Aumento de contador	contador <- contador + 1		17
Asignación de valor en el vector	numeros[contador] <- n		[16]=4
Llamada a la función ulam	n <- ulamRegla(n)		
Condición de residuo par igual a 0	m %% 2 == 0		
Retorna el valor a la mitad	return(m/2)	2	
Retorna el valor al triple más uno	return(3*m + 1)		
Condición de paro con n mayor a 1 y contador menor al límite	n > 1 & contador < limit		
Aumento de contador	contador <- contador + 1		18
Asignación de valor en el vector	numeros[contador] <- n		[16]=2
Llamada a la función ulam	n <- ulamRegla(n)		
Condición de residuo par igual a 0	m %% 2 == 0		
Retorna el valor a la mitad	return(m/2)	1	
Retorna el valor al triple más uno	return(3*m + 1)		
Encuentra el mínimo entre contador y limit	howMany <- min(contador, limit)		
Cuadro de texto para visualizar gráfica	show <- readline("Quieres visualizarla (y/n)? ")		y
Condición de valor en show igual a 'y'	show == "y"		
			7 22 11 34 17
			52 26 13 40 20
			10 5 16 8 4 2
Impresión de los números	print(numeros[1:howMany])		

Prueba de escritorio 1. Número= 88								
Descripción de sentencia	Sentencia	n	isValid	inicial	números []	contador	show	numeros
Lectura de numero	n <- suppressWarnings(as.integer(n))	88						
Validación de número y mayor a 1	isValid <- !is.na(n) && n > 1		true					
Declaración de inicial con valor n	inicial <- n			88				
Declaración de vector con límite de 1000	números <- numeric(limit=1000)				[]			
Declaración de contador con valor 0	contador <- 0					0		
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					1		
Asignación de valor en el vector	numeros[contador] <- n				[1]=88			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	44						
Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					2		
Asignación de valor en el vector	numeros[contador] <- n				[2] = 44			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	22						
Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					3		
Asignación de valor en el vector	numeros[contador] <- n				[3] = 22			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)							
Retorna el valor al triple más uno	return(3*m + 1)	11						
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					4		
Asignación de valor en el vector	numeros[contador] <- n				[4] = 11			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	34						
Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					5		
Asignación de valor en el vector	numeros[contador] <- n				[5] = 34			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)							
Retorna el valor al triple más uno	return(3*m + 1)	17						
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					6		
Asignación de valor en el vector	números[contador] <- n				[6] = 17			
Llamada a la función ulam	n <- ulamRegla(n)							
Condición de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	52						
Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					7		
Asignación de valor en el vector	números[contador] <- n				[7] = 52			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	26						

Retorna el valor al triple más uno	return(3*m + 1)						
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1				8		
Asignación de valor en el vector	numeros[contador] <- n			[8] = 26			
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)						
Retorna el valor al triple más uno	return(3*m + 1)	13					
Condición de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1				9		
Asignación de valor en el vector	numeros[contador] <- n			[9] = 13			
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	40					
Retorna el valor al triple más uno	return(3*m + 1)						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1				10		
Asignación de valor en el vector	numeros[contador] <- n			[10] = 40			
Llamada a la función ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	20					
Retorna el valor al triple más uno	return(3*m + 1)						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1				11		
Asignación de valor en el vector	numeros[contador] <- n			[11] = 20			
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	10					
Retorna el valor al triple más uno	return(3*m + 1)						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1				12		
Asignación de valor en el vector	numeros[contador] <- n			[12] = 10			
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)						
Retorna el valor al triple más uno	return(3*m + 1)	5					
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1				13		
Asignación de valor en el vector	numeros[contador] <- n			[13] = 5			
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	16					
Retorna el valor al triple más uno	return(3*m + 1)						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1				14		
Asignación de valor en el vector	numeros[contador] <- n			[14] = 16			
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	8					
Retorna el valor al triple más uno	return(3*m + 1)						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1				15		
Asignación de valor en el vector	numeros[contador] <- n			[15] = 8			
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	4					

Retorna el valor al triple más uno	return(3*m + 1)						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1					16	
Asignación de valor en el vector	numeros[contador] <- n				[16]=4		
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	2					
Retorna el valor al triple más uno	return(3*m + 1)						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1					17	
Asignación de valor en el vector	numeros[contador] <- n				[17]=2		
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)						
Retorna el valor al triple más uno	return(3*m + 1)						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1					18	
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	1					
Retorna el valor al triple más uno	return(3*m + 1)						
Encuentra el minimo entre contador y limit	howMany <- min(contador, limit)						
Cuadro de texto para visualizar grafica	show <- readline("Quieres visualizarla (y/n)? ")					y	
Condición de valor en show igual a 'y'	show == "y"						
							88 44 22
							11 34 17
							52 26 13
							40 20 10
Impresión de los numeros	print(numeros[1:howMany])						5 16 8 4 2

Prueba de escritorio 1. Número = 7

Descripción de sentencia	Sentencia	n	isValid	inicial	numeros[]	contador	show	numeros
Lectura de numero	n <- suppressWarnings(as.integer(n))	7						
Validación de numero y mayor a 1	isValid <- !is.na(n) && n > 1		true					
Declaracion de inicial con valor n	inicial <- n			7				
Declaracion de vector con limite de 1000	numeros <- numeric(limit=1000)				[]			
Declaración de contador con valor 0	contador <- 0					0		
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					1		
Asignación de valor en el vector	numeros[contador] <- n				[1]=7			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)							
Retorna el valor al triple más uno	return(3*m + 1)	22						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					2		
Asignación de valor en el vector	numeros[contador] <- n				[2] = 22			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	11						
Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							

Aumento de contador	contador <- contador + 1					3		
Asignación de valor en el vector	numeros[contador] <- n				[3] = 11			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)							
Retorna el valor al triple más uno	return(3*m + 1)	34						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					4		
Asignación de valor en el vector	numeros[contador] <- n				[4] = 34			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	17						
Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					5		
Asignación de valor en el vector	numeros[contador] <- n				[5] = 17			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)							
Retorna el valor al triple más uno	return(3*m + 1)	52						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					6		

Asignación de valor en el vector	numeros[contador] <- n				[6] = 52			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	26						
Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					7		
Asignación de valor en el vector	numeros[contador] <- n				[7] = 26			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	13						
Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					8		
Asignación de valor en el vector	numeros[contador] <- n				[8] = 13			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)							
Retorna el valor al triple más uno	return(3*m + 1)	40						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					9		
Asignación de valor en el vector	numeros[contador] <- n				[9] = 40			

Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	20						
Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					10		
Asignación de valor en el vector	numeros[conta dor] <- n				[10] = 20			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	10						
Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					11		
Asignación de valor en el vector	numeros[conta dor] <- n				[11] = 10			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	5						
Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					12		
Asignación de valor en el vector	numeros[conta dor] <- n				[12] = 5			
Llamada a la funcion ulam	n <- ulamRegla(n)							

Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)						
Retorna el valor al triple más uno	return(3*m + 1)	16					
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1				13		
Asignación de valor en el vector	numeros[contador] <- n				[13] = 16		
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	8					
Retorna el valor al triple más uno	return(3*m + 1)						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1				14		
Asignación de valor en el vector	numeros[contador] <- n				[14] = 8		
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	4					
Retorna el valor al triple más uno	return(3*m + 1)						
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit						
Aumento de contador	contador <- contador + 1				15		
Asignación de valor en el vector	numeros[contador] <- n				[15] = 4		
Llamada a la funcion ulam	n <- ulamRegla(n)						
Condicion de residuo par igual a 0	m %% 2 == 0						
Retorna el valor a la mitad	return(m/2)	2					

Retorna el valor al triple más uno	return(3*m + 1)							
Condicion de paro con n mayor a 1 y contador menor al limite	n > 1 & contador < limit							
Aumento de contador	contador <- contador + 1					16		
Asignación de valor en el vector	numeros[contador] <- n				[16]=2			
Llamada a la funcion ulam	n <- ulamRegla(n)							
Condicion de residuo par igual a 0	m %% 2 == 0							
Retorna el valor a la mitad	return(m/2)	1						
Retorna el valor al triple más uno	return(3*m + 1)							
Encuentra el minimo entre contador y limit	howMany <- min(contador, limit)							
Cuadro de texto para visualizar grafica	show <- readline("Quieres visualizarla (y/n)? ")							
Condición de valor en show igual a 'y'	show == "y"						y	
Impresión de los numeros	print(numeros[1:howMany])							7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2

Conclusión

Se demostró que existen operaciones aplicables a cualquier entero positivo, sea número par o número impar. Donde para los números pares $f(n) = \frac{n}{2}$ y los números impares $f(n) = 3n + 1$. Debido a lo anterior se supo demostrar que no existen ciclos cerrados o secuencias divergentes de números impares que no cumplan la conjetura, por lo tanto, el proceso de multiplicar por 3 cualquier número impar y al resultado sumarle uno, no puede contener ciclos cerrados, ni el proceso es infinito.