Pregunta 1

```
============================================================
GRAMÁTICA: roman_parser1
============================================================

CONJUNTOS FIRST:
------------------------------------------------------------
FIRST(roman):        ['C', 'D', 'I', 'L', 'V', 'X', 'λ']
FIRST(hundreds):     ['C', 'D', 'λ']
FIRST(low_hundreds): ['λ']
FIRST(tens):         ['L', 'X', 'λ']
FIRST(low_tens):     ['λ']
FIRST(units):        ['I', 'V', 'λ']
FIRST(low_units):    ['λ']

CONJUNTOS FOLLOW:
------------------------------------------------------------
FOLLOW(roman):       ['$']
FOLLOW(hundreds):    ['$', 'I', 'L', 'V', 'X']
FOLLOW(low_hundreds): ['$', 'C', 'I', 'L', 'V', 'X']
FOLLOW(tens):        ['$', 'I', 'V']
FOLLOW(low_tens):    ['$', 'I', 'V', 'X']
FOLLOW(units):       ['$']
FOLLOW(low_units):   ['$', 'I']


============================================================
GRAMÁTICA: roman_parser2
============================================================

CONJUNTOS FIRST:
------------------------------------------------------------
FIRST(roman):        ['C', 'D', 'I', 'L', 'V', 'X', 'λ']
FIRST(hundreds):     ['C', 'D', 'λ']
FIRST(low_hundreds): ['C', 'λ']
FIRST(tens):         ['L', 'X', 'λ']
FIRST(low_tens):     ['X', 'λ']
FIRST(units):        ['I', 'V', 'λ']
FIRST(low_units):    ['I', 'λ']

CONJUNTOS FOLLOW:
------------------------------------------------------------
FOLLOW(roman):       ['$']
FOLLOW(hundreds):    ['$', 'I', 'L', 'V', 'X']
FOLLOW(low_hundreds): ['$', 'I', 'L', 'V', 'X']
```

FOLLOW(tens): ['$', 'I', 'V']
FOLLOW(low_tens): ['$', 'I', 'V']
FOLLOW(units): ['$']
FOLLOW(low_units): ['$']

==
DIFERENCIAS ENTRE LAS GRAMÁTICAS:
==

1. DIFERENCIAS EN CONJUNTOS FIRST:

FIRST(low_hundreds):
  G1: ['λ']
  G2: ['C', 'λ']
  Solo en G2: ['C']

FIRST(low_tens):
  G1: ['λ']
  G2: ['X', 'λ']
  Solo en G2: ['X']

FIRST(low_units):
  G1: ['λ']
  G2: ['I', 'λ']
  Solo en G2: ['I']

2. DIFERENCIAS EN CONJUNTOS FOLLOW:

FOLLOW(low_hundreds):
  G1: ['$', 'C', 'I', 'L', 'V', 'X']
  G2: ['$', 'I', 'L', 'V', 'X']
  Solo en G1: ['C']

FOLLOW(low_tens):
  G1: ['$', 'I', 'V', 'X']
  G2: ['$', 'I', 'V']
  Solo en G1: ['X']

FOLLOW(low_units):
  G1: ['$', 'I']
  G2: ['$']
  Solo en G1: ['I']

Pregunta 2

```
========================================
1. GRAMÁTICA 1 (Reglas de producción)
========================================
S' -> R
R  -> H T U
H  -> Lh | C D | D Lh | C M
Lh -> Lh C | lambda
T  -> Lt | X L | L Lt | X C
Lt -> Lt X | lambda
U  -> Lu | I V | V Lu | I X
Lu -> Lu I | lambda

(Nota: 'lambda' representa la cadena vacía)

========================================
2. ESTADOS Y CONJUNTOS DE ÍTEMS
========================================

--- ESTADO S0 ---
S' -> .R       , $
R  -> .H T U   , $
H  -> .Lh      , L/X/I/V/$
H  -> .C D     , L/X/I/V/$
H  -> .D Lh    , L/X/I/V/$
H  -> .C M     , L/X/I/V/$
Lh -> .Lh C    , L/X/I/V/$
Lh -> .        , C/L/X/I/V/$

--- ESTADO S1 (Transición con R) ---
S' -> R.       , $

--- ESTADO S2 (Transición con H) ---
R  -> H.T U    , $
T  -> .Lt      , I/V/$
T  -> .X L     , I/V/$
T  -> .L Lt    , I/V/$
T  -> .X C     , I/V/$
Lt -> .Lt X    , I/V/$
Lt -> .        , X/I/V/$

--- ESTADO S3 (Transición con Lh) ---
H  -> Lh.      , L/X/I/V/$
Lh -> Lh.C     , L/X/I/V/$

--- ESTADO S4 (Transición con C) ---
```

H -> C.D   , L/X/I/V/$
H -> C.M   , L/X/I/V/$

--- ESTADO S5 (Transición con D) ---
H -> D.Lh   , L/X/I/V/$
Lh -> .Lh C   , L/X/I/V/$
Lh -> .   , C/L/X/I/V/$

==========
1. GRAMÁTICA 2 (Reglas de producción)
==========
S' -> R
R -> H T U
H -> Lh | C D | D Lh | C M
Lh -> C Lh | lambda
T -> Lt | X L | L Lt | X C
Lt -> X Lt | lambda
U -> Lu | I V | V Lu | I X
Lu -> I Lu | lambda

(Nota: 'lambda' representa la cadena vacía)

==========
2. ESTADOS Y CONJUNTOS DE ÍTEMS
==========

--- ESTADO S0 ---
S' -> .R   , $
R -> .H T U   , $
H -> .Lh   , L/X/I/V/$
H -> .C D   , L/X/I/V/$
H -> .D Lh   , L/X/I/V/$
H -> .C M   , L/X/I/V/$
Lh -> .C Lh   , L/X/I/V/$
Lh -> .   , L/X/I/V/$

--- ESTADO S1 (Transición con R) ---
S' -> R.   , $

--- ESTADO S2 (Transición con H) ---

R -> H.T U   , $
T -> .Lt   , I/V/$
T -> .X L   , I/V/$
T -> .L Lt   , I/V/$
T -> .X C   , I/V/$
Lt -> .X Lt   , I/V/$
Lt -> .   , I/V/$

--- ESTADO S3 (Transición con Lh) ---
H -> Lh.   , L/X/I/V/$

--- ESTADO S4 (Transición con C) ---
H -> C.D   , L/X/I/V/$
H -> C.M   , L/X/I/V/$
Lh -> C.Lh   , L/X/I/V/$
Lh -> .C Lh   , L/X/I/V/$
Lh -> .   , L/X/I/V/$

--- ESTADO S5 (Transición con D) ---
H -> D.Lh   , L/X/I/V/$
Lh -> .C Lh   , L/X/I/V/$
Lh -> .   , L/X/I/V/$

1. DIFERENCIAS EN ITEMS DE ESTADO S0:

G1:
Lh -> .   , C/L/X/I/V/$
Símbolo de adelanto C solo en G1 (provocará conflicto)

G2:
Lh -> .   , L/X/I/V/$

Pregunta 3

TABLA DE ANÁLISIS LR(1) - GRAMÁTICA G1 (Roman Parser 1)
==========

	ACCIÓN (TERMINALES)							IR A (NO TERMINALES)				
ESTADO	C	D	M	L	X	I	V	$	R	H	T	Lh	Lt
S0	S4 / r7 (CONFLICT)	S5		r7	r7	r7	r7	r7	S1	S2		S3	
S1								acc					
S2				Shift	r13	r13	r13	r13			Goto		Goto
S3	Shift			r2	r2	r2	r2	r2					
S4		Shift	Shift										
S5	r7			r7	r7	r7	r7	r7				Goto	

DETALLE DE CONFLICTOS ENCONTRADOS (coincide con lo que aparece en parser.out):

1. ESTADO S0 (Entrada 'C'), conflicto shift/reduce:
   - Shift S4: Generado por H -> .C D, L/X/I/V/$ (y H -> .C M, L/X/I/V/$)
   - Reduce r7: Generado por Lh -> . , C/L/X/I/V/$

Es gramática LR1?
Al haber conflictos en la tabla, no es una gramática LR1

Pregunta 4

Hacer recursividad por la izquierda (como en la G1) provoca que al hacer los items de los estados LR1, se expanda una vez más el item de Lh a partir del siguiente item:
Lh -> .Lh C , L/X/I/V/$
Esto provoca que se añada C a los símbolos de adelanto, lo que provoca un conflicto.