

UTN FRBA – SSL – Examen Final – 2025-09-25

Apellido, Nombre:	Legajo:	Nota:
-------------------	---------	-------



- Resuelva el examen en tinta y en esta hoja; no se aceptan hojas adicionales.
- Para los ítems de *una mejor respuesta*, marcados con una círculo (○), tilde (✓) sólo una opción, la mejor.
- Para los ítems de *respuestas múltiple*, marcados con un caja (□), tilde (✓) todas las respuestas correctas.
- Durante el examen no se responde consultas; si lo necesita, escriba hipótesis de trabajo, las cuales también se evalúan.

1. (2 puntos) Dada la declaración `int x=0, a=2;`, evalúe cada expresión, indique su valor y el valor resultante de la variable `a`.

Expresión	Valor de la expresión	Valor resultante de <code>a</code>
<code>x && ++a</code>		
<code>x < 'a' ? x + 'a' : a += 40</code>		

2. (1 punto) Analice las siguientes ERX ¿cuál representa un *sublenguaje* del LF *identificadores de C*?

- ☐ `a`
- ☐ `[a-c0-2]`
- ☐ `[0-2a-c]`
- ☐ `[a-cA-C_]*`
- ☐ `[a-cA-C0-9]`

3. (2 puntos) Defina formalmente el AF que resulta de la intersección entre las constantes octales sin sufijo de C, las expresiones de C y el LF "números sin signo expresados en base dos".

4. Analice la siguiente expresión C: `lim(x --> 0)`

- (2 puntos) Si es sintácticamente correcta entonces escriba **una** declaración que lo haga también semánticamente correcto, si no, justifique.
- (2 puntos) ¿La expresión calcula el límite de una función cuando `x` tiende a 0? Justifique.
- (2 puntos) ¿Agregar o quitar **un** espacio en algún lugar podría cambiar la cantidad de lexemas? Justifique.

1. Una Resolución

1.

Expresión	Valor de la expresión	Valor resultante de a
x && ++a	0	2
x < 'a' ? x + 'a' : a += 40	'a'	2

2. a

3. $M = (\{A, B\}, \{0, 1\}, \{A \Rightarrow 0 \Rightarrow B, B \Rightarrow 0 \Rightarrow B, B \Rightarrow 1 \Rightarrow B\}, 0, \{B\})$

4. a. `int x, lim(bool);`

b. No, `-->` no es un operador, son dos: posdecremento y mayor, no representa la semántica de *tiende a*.

c. • Si no agregamos espacios, son siete lexemas:

`lim, (, x, --, >, 0,)`

• Si agregamos un espacio después del primer `-`, siguen siendo siete lexemas:

`lim, (, x, -, -->, 0,)`

• Si agregamos **dos** espacios, uno después de cada `-`, son ocho lexemas

`lim, (, y, -, -, >, 0,)`

• Pero, si agregamos un espacio después después de la `l`, son ocho lexemas:

`l, im, (, x, -->, 0,)`

• Lo mismo, si agregamos un espacio después de la `i`:

`li, m, (, x, -->, 0,)`

v1.0.0-beta.1+2025-09-26