

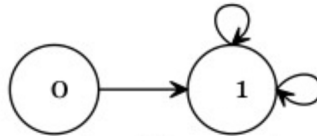
1. UTN FRBA – SSL – K2003 – Examen #2 – 2025-11-05 – Tema A

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 un 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. (5 puntos) Complete la especificación del autómata finito para que reconozca *identificadores* de C.



- Alfabeto:
 - Etiquetas de las transiciones y estados:
 - Dibuje el estado de rechazo.
 - Indique la máxima longitud que puede tomar una palabra.
 - Indique la mínima longitud que puede tomar una palabra.
2. (4 puntos) Dada la declaración `int getchar()`, `putchar()`, `*p`; indique con una cruz que tipo de comportamiento tiene cada expresión:

Expresión	Definido	Indefinido	No especificado	Especificado por la implementación	Específico de locación
<code>putchar('A')+putchar('B')</code>					
<code>*p=getchar()</code>					
<code>p=getchar()</code>					
<code>sizeof *p > 6</code>					

3. (6 puntos) Traduzca al castellano las siguientes declaraciones:

- `T *m[2][3];`
- `T (*pa)[4];`
- `T *fp();`

4. (4 puntos) Escriba un fragmento de programa que declare un objeto automático que apunte a un objeto estático:

5. (15 puntos) Indique la visibilidad o alcance, tiempo de vida, y espacio de nombre de los identificadores del siguiente programa:

```
/*1*/ struct s{int x;} v;
/*2*/
/*3*/ void g(){
/*4*/ static int a;
/*5*/ }
```

	Visibilidad	Tiempo de vida	Espacio de nombre
s			
x			
v			
g			
a			

6. (6 puntos) Agregue paréntesis redundantes a estas expresiones para denotar asociatividad y precedencia:

1. a = b = c
2. & a . b . c
3. a == b == c

7. Responda las siguientes preguntas en el contexto del trabajo *traductor de declaraciones C a castellano*

- a. (2 puntos) Tilde las categorías léxicas que se aplican:

- ☐ expresión.
- ☐ declarador.
- ☐ identificador.
- ☐ palabra clave.
- ☐ llave-izquierda.
- ☐ corchete-derecho.

- b. (2 puntos) Tilde las categorías sintácticas que se aplican:

- ☐ constante.
- ☐ sentencia.
- ☐ declaración.
- ☐ punto-y-coma.
- ☐ declarador-directo
- ☐ unidad-de-traducción.

- c. Suponga que traductor recibe por *stdin* el contenido del siguiente archivo:

```
int int;
```

- i. (2 puntos) Indique si tiene errores léxicos, sintácticos, o semánticos. Justifique.
- ii. (1 punto) ¿Cuántos invocaciones a `ungetc` necesita hacer el Scanner?

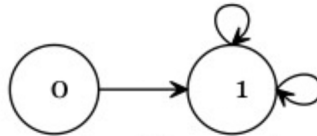
2. UTN FRBA – SSL – K2002 – Examen #2 – 2025-11-04 – Tema A

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. (5 puntos) Complete la especificación del autómata finito para que reconozca *identificadores* de C.



- Alfabeto:
- Etiquetas de las transiciones y estados:
- Indique cuantas transiciones son necesarias para rechazar o aceptar estas cadenas, e indique si las acepta o rechaza:
 - 7ab
 - for
 - abc

2. (5 puntos) Dada la declaración `int c=getchar()`, `*p`; indique con una cruz que tipo de comportamiento tiene cada expresión:

Expresión	Definido	Indefinido	No especificado	Especificado por la implementación	Específico de locación
<code>islower(c)</code>					
<code>islower(c)==islower(c)</code>					
<code>islower(*p)</code>					
<code>sizeof *p</code>					
<code>&c != &p</code>					

3. (10 puntos) Traduzca al castellano las siguientes declaraciones:

- `int *x[7];`
- `int (*x)[7];`
- `int *x();`
- `int (*x)();`
- `int (*x[7])();`

4. (4 puntos) Declare un objeto con clase de almacenamiento estática que apunte a un objeto con almacenamiento *alocado*:

5. (15 puntos) Indique la visibilidad, tiempo de vida, y espacio de nombre de los identificadores del siguiente programa:

```

/*1*/ int a;
/*2*/
/*3*/ void f(){
/*4*/   int b;
/*5*/ }
/*6*/
/*7*/ void g(){
/*8*/   char a;
/*9*/ }

```

	Visibilidad	Tiempo de vida	Espacio de nombre
a (<i>int</i>)			
f			
b			
g			
a (<i>char</i>)			

6. (6 puntos) Agregue paréntesis redundantes a estas expresiones para denotar asociatividad y precedencia:

1. a = b = c
2. a + b + c
3. * a . b [c]

7. (6 puntos) En el contexto del trabajo *traductor de declaraciones C a castellano*, suponga que traductor recibe por stdin el contenido del siguiente archivo:

```
char c1;
```

- a. ¿Cuántas invocaciones a `GetNextToken` necesita hacer el Parser?
- b. ¿Cuántas invocaciones a `getchar` necesita hacer el Scanner?
- c. ¿Cuántas invocaciones a `ungetc` necesita hacer el Scanner?

v2.1.0-rc.5, 2025-11-07