

UTN FRBA – SSL – Examen Final – 2020-12-14

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



- Resuelva el examen en este documento; no se aceptan documentos adicionales.
- Durante el examen no se responde consultas; si lo necesita, escriba hipótesis de trabajo, las cuales también se evalúan.

1. Dado el fragmento: `wiehl()--x;`

- (1 punto) Enumere en orden los caracteres que son devueltos al flujo mediante `ungetc` durante el análisis léxico:
- (2 puntos) Realice un análisis de sintáctico de **izquierda a derecha** y **justifique** si es una *sentencia* sintácticamente correcta. Si hay error sintáctico, reescribala con el error encontrado corregido:
- (2 puntos) Escriba una declaración que haga la *sentencia* anterior, ya sea la original o la corregida, **semánticamente** correcta:

2. Dada la siguiente función:

```
int f(int c){
    int d = c-7;
    printf("%d",d);
    return d;
}
```

- (2 puntos) Analice sintácticamente el cuerpo de la función e indique:
 - Cantidad de sentencias:
 - Cantidad de expresiones completas:
 - Cantidad de declaraciones:
- (3 puntos) Analice las siguientes expresiones. Indique su valor y su salida por `stdout`, si la salida no se puede determinar, indique "*Indeterminada*":

Expresión	Valor	Salida por <code>stdout</code> o " <i>Indeterminada</i> "
<code>f(7)-f(7)</code>		
<code>f(8) f(7)</code>		
<code>f(7) f(8)</code>		

3. (Punto extra) ¿Explique para qué usaría la keyword `union` en la construcción de *scanners*?

1. Una Resolución

1. a. (,;
- b. El análisis sintáctico de izquierda a derecha reconoce como una expresión correcta a `weihl()`, ya que es una expresión sufijo, más precisamente una de invocación con lista de argumentos vacía. El problema aparece luego de `)`, no se espera un operador prefijo; hay varios tokens que pueden seguir a la invocación, pero no un operador prefijo no es uno de ellos.
Algunas soluciones para corregir el error encontrado son:

i. `weihl()-x;`

ii. `weihl()*--x;`

c. `int weihl(void),x;`

d. `int weihl(void),*x;`

2. a. i. Cantidad de sentencias: 2
ii. Cantidad de expresiones completas: 3
iii. Cantidad de declaraciones: 1

b.

0	00
1	1
1	01

3. Para modelar el valor semántico del token.

v1.1.0, 2020-12-21