

UTN FRBA – SSL – Examen Final – 2019-12-18

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. Considerando `int f(int),g(int);` con `f` y `g` definidas, analice la siguiente función:

```
/*1*/ int h(int x){  
/*2*/   int y=0;  
/*3*/   y=f(x);  
/*4*/   y=g(y);  
/*5*/   return y;}
```

- (1 punto) Indique cuantas sentencias tienen la sentencia compuesta de la función `h`.
 - Analice la sentencia de la línea 3:
 - (1 punto) Enumere cada operación realizada e indique la precedencia, uno es la menor.
 - (1 punto) Escriba en renglones diferentes las subexpresiones que tiene la expresión.
 - (Punto extra) Dibuje el árbol asociado a la expresión.
 - El compilador emite en la línea tres el mensaje: “se asigna un valor que no es utilizado”.
 - (2 puntos) Si coincide con el mensaje categorice el tipo de error. Justifique.
 - (2 puntos) Proponga una solución para evitar el error. Justifique.
2. (1 punto) Indique el orden de las etapas fuera y dentro del proceso de compilación, donde uno (1) es la primera. Si no corresponde indíquelo con un guión:

- ☐ Vinculación.
- ☐ Análisis léxico.
- ☐ Previnculación.
- ☐ Análisis sintáctico.
- ☐ Preprocesamiento.
- ☐ Análisis semántico.
- ☐ Generación de código.

3. (2 puntos) Tilde todas las afirmaciones **verdaderas** acerca de C:

- ☐ Una definición de variable implica reserva de memoria.
- ☐ Una declaración de variable implica reserva de memoria.
- ☐ Las variables `static` (estáticas) de tipo `int` se inicializan por defecto en cero cuando no tienen inicializador explícito.
- ☐ Las variables `static` (estáticas) definidas externamente (fuera de toda función) siempre son accesibles desde toda la UT.
- ☐ Para compilar con éxito una unidad de traducción (UT) debe tener las definiciones de todos los identificadores que aparecen en la misma.

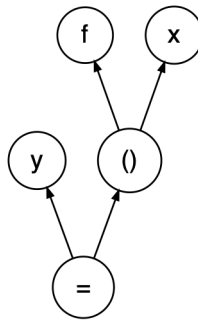
1. Una Resolución

1. a. 3

b. i. =, ().

- ii.
- y
 - f
 - x
 - f(x)

iii.



c. i. Warning, posible error pragmático.

El compilador detecta en la línea 3 que el valor previamente asignado a y, en particular en la declaración de la línea 2, no es utilizado ya que y se sobrescribe con el valor retornado por f(x), y el 0 con el cual se había inicializado y no tiene uso en el programa. La inicialización tiene el efecto de asignación.

No hay error semántico porque el compilador entiende el significado que expresamos. Es tan solo un *posible* error pragmático porque sí hay significado, y la inicialización innecesaria de la línea 3 puede ser, simplemente, un estilo de codificación aplicado por el programador.

ii. `int h(int x){return g(f(x));}`.

Las asignaciones temporales a y son innecesarias no hay efecto de lado fuera de h, el mismo efecto se logra en una única sentencia. El comportamiento que expone la función es la de composición de la funciones g con f.

2. 6, 2, -, 3, 1, 4, 5

3. ✓

□

✓

□

□

v1.0.1, 2020-03-03