



## Ejercicios de Final

1. Sea `int a=4,b=2`; compare `a=a+b` y `a+=b` e indique cuál afirmación es falsa:
  - ☐ Son expresiones.
  - ☐ Tienen mismo valor.
  - ☐ Tienen mismo tipo de dato.
  - ☐ Tienen mismo efecto de lado.
  - ☒ **Tienen misma cantidad de evaluaciones.**
2. Sea `double d=1`; analice el fragmento `++d++` y responda:
  - a. ¿A qué categoría sintáctica pertenece?: **Expresión**
  - b. ¿Cuántas operandos tiene?: **2**
  - c. ¿Es semánticamente correcta? ¿Por qué?: **No, porque d++ NO es un valorL modificable**
3. Dadas las siguientes funciones, indique el valor de cada expresión y, si es que está determinada, la salida por stdout, si no la frase "Indeterminada":
  - `int g(void){putchar('g');return 'g'-'f';}`
  - `int f(void){putchar('f');return '\0';}`

Expresión	Valor	Salida por stdout o "Indeterminada"
<code>g()&amp;&amp;f()</code>	0	gf
<code>g()*f()</code>	0	Indeterminada
<code>g()  f()</code>	1	g

4. Analice sintácticamente la siguiente función y describa tres errores sintácticos, en orden de aparición:
  - `void f(int a){int a=(); switch(a){a} return 2;}}`
  - a. Error 1: **Los paréntesis requieren una expresión.**
  - b. Error 2: **La sentencia expresión requiere un punto y coma.**
  - c. Error 3: **No se espera el token llave que cierra.**
5. Analice la siguiente sentencia compuesta:
  - `{short a=0;{char a;a;}while(1)a;}`
  - a. ¿Cuántos lexemas posee?: **20**
  - b. ¿Cuántas expresiones?: **4**
  - c. ¿Es semánticamente correcta? ¿Por qué?: **Sí, es semánticamente correcta porque la segunda variable a oculta la primera. Además, ambas sentencias expresión a; no tienen efecto de lado, y la iteración es infinita.**



6. Dada la siguiente expresión sintácticamente correcta: `a[i]=*p`  
escriba tres declaraciones para esa expresión, una que la haga semántica correcta, otra semánticamente incorrecta por Lvalue, y otra semánticamente incorrecta por operación y operador incompatible.
- Declaración que hace a la expresión semánticamente correcta:  
**`int a[1]={0}, i=0, *p=a;`**
  - Declaración que hace a la expresión semánticamente incorrecta por error asociado a Lvalue:  
**`const int a[1]={0}, i=0, *p=a;`**
  - Declaración que hace a la expresión semánticamente incorrecta por error asociado a operador y operando incompatibles:  
**`int a[1]={0}, i=0, p=1;`**
7. Indique el concepto que no está asociado a la sintaxis de las Expresiones de C, y justifique:
- ☐ Precedencia.
  - ☐ Asociatividad.
  - ☒ Efecto de lado.
  - ☐ Árbol de derivación.
  - ☐ Invocación de función.

**Justificación: El efecto de lado no se define en la sintaxis de las expresiones de C, es un concepto semántico que indica el comportamiento al evaluar una expresión. Por otro lado, la precedencia y la asociatividad sí se definen en la sintaxis, y el árbol de derivación de una expresión se arman en función de la sintaxis.**

8. Dado el fragmento: `fi(x>0)a=x;`
- Enumere en orden los caracteres que son devueltos al flujo mediante `ungetc` durante el análisis léxico: **`(>0)=x;`**
  - 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:  
**El análisis sintáctico de izquierda a derecha reconoce como una expresión correcta a `fi(x>0)`, ya que es una expresión sufijo, más precisamente una de invocación, cuyo argumento es `x>0` que es una expresión relacional correcta. El problema aparece luego de `)`, no se espera un identificador; hay varios tokens que pueden seguir a la invocación, pero un identificador no es uno de ellos.**

Algunas soluciones para corregir el error encontrado son:

- `fi(x>0);a=x; // dos sentencias.`
- `fi(x>0),a=x; // una sentencia con dos expresiones.`
- `*(fi(x>0)+a)=x; // una sentencia expresión con una asignación, el lvalue es un offset a desde una base retornada por la función fi.`



- c. Escriba una declaración que haga la sentencia anterior semánticamente correcta:  
Algunas declaraciones posibles, que se correlacionan a las correcciones anteriores:
- i. `int fi(int),x,a;`
  - ii. `char fi(int),x,a;`
  - iii. `int *fi(int),x,a; // fi retorna un puntero a int.`