Ejercicios de Sintaxis

Nota: los ejercicios marcados con (*) al principio están sacados del libro de la cátedra
los ejercicios marcados con (°) al principio están basados en uno tomado en un final
los ejercicios marcados con (^) al principio están basados en uno tomado en un final (reciente)

1. Dado el siguiente fragmento de código ANSI C liste en la tabla los lexemas que reconoce el escáner e indique a que categoría pertenecen.

```
int i;
double v[10] , *p;
/* fin declaraciones, inicio sentencias */
for (i = 0; i < 10; i++) {
      p = una_funcion(i);
      v[i] = *p;
}
```

lexema	token	lexema	token	lexema	token

2. (°) Sea el siguiente fuente en ANSI C:

```
#define a 10
b/*e*/c/**/da:a=b/***/*/10"\*a*/"
```

Escriba en la tabla de abajo los lexemas detectados por el escáner y a que categoría pertenecen, suponiendo que estas son: IDENTIFICADOR, CONSTANTE y OTRO

LEXEMA	CATEGORÍA LÉXICA

3. (°) Sean las siguientes categorías léxicas:
TOKEN → <palRes> | <ident> | <otro> | <constReal>
Realice el análisis léxico de la siguiente función ANSI C:
INT main printf) { ++; 2.3E-8; }

LEXEMA	CATEGORÍA LÉXICA

4. (°) Sea la siguiente sentencia compuesta ANSI C

```
{ int i=0; int c;
  for ( ; isdigit(c); i++) ;
  ungetc(c, stdin); }
```

Determine

a) Cantidad de lexemas que hay DENTRO de la sentencia compuesta:
b) Cantidad de sentencias que hay DENTRO de esta sentencia compuesta:

5. (°) Sean las categorías gramaticales (o sintácticas) **Expresiones**, **Declaraciones** y **Sentencias** de ANSI C. Indique que categoría, según el compilador, le corresponde a cada uno de los siguientes constructos:

CONSTRUCTO	CATEGORÍA
`{printf;}`	
`main = scanf`	
`struct punto {double x; double y;} punto;`	
`int a, x(void);`	
`a+++++b;`	
`while(1);`	

6. (°) Indique el tipo de dato (en ANSI C) de cada una de las siguientes expresiones o escriba ERROR si no es una expresión

EXPRESIÓN	TIPO DE DATO
`-0.266`	
`14L`	
`22.6 > 4`	
`'a' + 22.F`	
`12.8 && -12.8`	

7. (°) Por cada cadena de la siguiente tabla indique a que CATEGORÍA DE CONSTANTE pertenece el lexema asociado. Las categorías son: entera decimal, entera hexadecimal, entera octal, real, carácter. Además escriba el TIPO DE DATO de cada constante. Si no es posible formar una constante de las categorías dadas, escriba ERROR.

Cadena	Categoría de Constante	Tipo de dato
`7f`		
`'\\'`		
`10L`		
`“C”`		
`4.3e8`		
`0127`		

8. (°) Marque con una cruz si los siguientes constructos ANSI C tiene errores semánticos, errores sintácticos, o no tiene error de compilación. Asuma que las funciones estándar están disponibles.

	Errores Semánticos	Errores Sintácticos	Sin Error
`{int main=0; printf(“%d\n”, main);}`			
`{char a[5]; a[15]='A'+2;}`			
`{int a=0,b; {while (a<10) {++a; b=b+2;}}`			
`{int p=5; for ( ;5; ) 8 = p-- ;}`			

9. Marque con una cruz si los siguientes constructos ANSI C tiene errores semánticos, errores sintácticos, o no tiene error de compilación. Asuma que las funciones estándar están disponibles.

	Errores Semánticos	Errores Sintácticos	Error Léxico	Sin Error
`{char a='a'; for(;;) 'F'=a++;}`				
`{char a[10]; a[-2]='B'+2;}`				
`{int x=0,y=1,z; if x<y z=x+y;}`				
`{int a=0; a @= 1;}`				
`{int i,j=5; void *p; p=&j; i=*p;}`				

10. Marque con una cruz si los siguientes constructos ANSI C tiene errores semánticos, errores sintácticos, o no tiene error de compilación. Asuma que las funciones estándar están disponibles.

	Errores Semánticos	Errores Sintácticos	Error Léxico	Sin Error
`{int 2z=0; 2z += 1;}`				
`{int a=7; while a-- printf("%d", a);}`				
`{char *a = malloc(5); a[8]='A'+2;}`				
`{int a; float a; for(;;) a++;}`				
`{int v[5],*p; p=v; *v++=3; *p++=2;}`				

11. (^) 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.

12. (^) Sea double `d=1;` analice el fragmento `++d++` y responda:
    a) ¿A qué categoría sintáctica pertenece?:
    b) ¿Cuántos operandos tiene?:
    c) ¿Es semánticamente correcta? ¿Por qué?:

13. (^) 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"
`g()&&f()`		
`g()*f()`		
`g()||f()`		

14. (^) 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:

    b) Error 2:

    c) Error 3:

15. (^) Analice la siguiente sentencia compuesta:
    - `{short a=0;{char a;a;}while(1)a;}`

    a) ¿Cuántos lexemas posee?:

    b) ¿Cuántas expresiones?:

    c) ¿Es semánticamente correcta? ¿Por qué?:

16. (^) 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.

    a) Declaración que hace a la expresión semánticamente correcta:

    b) Declaración que hace a la expresión semánticamente incorrecta por error asociado a Lvalue:

    c) Declaración que hace a la expresión semánticamente incorrecta por error asociado a operador y operando incompatibles:

17. (^) 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:

18. (^) Dado el fragmento: `fi(x>0)a=x;`

    a) Enumere en orden los caracteres que son devueltos al flujo mediante ungetc durante el análisis léxico:

    b) Realice un análisis de sintáctico de izquierda a derecha y justifique si es una sentencia sintácticamente correcta. Si hay error sintáctico, reescríbala con el error encontrado corregido:

    c) Escriba una declaración que haga la sentencia anterior semánticamente correcta:

19. (^) Sea `char*v="SSL";` indique con un tilde todas las expresiones que sí son ValorL:
    - ❑ `v`
    - ❑ `*v`
    - ❑ `v[3]`
    - ❑ `*(v+3)`

    Adicional:

    a) Cuales son valores L modificables?

    b) Que cambia si el código analizado fuese: `char v[]="SSL";`

20. (^) 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 a
`x && ++a`		
`x<'a' ? x+'a' : a+=40`		

21. (^) Dada la gramática
```
sentencia-for:
 for ( expresión ; expresión ; expresión ) { sentencia }
```
¿Representa las sentencias-for de C? (Justifique)

22. (^) Analice léxicamente el string `x>>>>>y`, ¿cuántos tokens posee?
    - ❑ Cinco.
    - ❑ Cantidad indeterminada porque existe ambigüedad.
    - ❑ Cantidad indeterminada porque existe error léxico.
    - ❑ Para el análisis léxico se requieren espacios.
    - ❑ Siete.

23. (^) Escriba un ejemplo de una expresión C con por lo menos un error semántico. Justifique.

24. (^) Sea la expresión `++automata->finales[i]`:

    a) Reescriba la expresión con paréntesis redundantes que expliciten la precedencia y la asociatividad:

    b) Escriba las declaraciones para que sea semánticamente correcta.

25. (^) Escriba declaraciones que hagan que la siguiente expresión sea semánticamente válida y que la expresión a la derecha de la asignación sea un valor-l no modificable.
```
c = a[1].b
```

26. (^) Declare v para que la siguiente expresión sea semánticamente correcta: `v[3]()`