

Apellido y Nombre \_\_\_\_\_

Legajo \_\_\_\_\_

- El examen debe resolverse en tinta y en esta hoja; no se aceptan hojas adicionales.
- Durante el examen no se responde consultas; de ser necesario, escriba hipótesis de trabajo, las cuales también se evalúan.

**Parte 1 - Ejercicios**

1 Justifique el valor de verdad de la siguiente afirmación:

La declaración `int T(int,int);` permite implementar la función transición de un AFPD.

2 Dado  $\Sigma=\{a,b\}$ , siendo L1 el LF de las palabras que terminan en a, y L2 el de las palabras que comienzan en a, complete la función para que retorne si s es una palabra de  $L3=L1 \cap L2$

```

EsPalabraDeL3(const _____ *s){
    if(*s++ != _____)
        return 0;
    while(_____ '\0'){
        if( * _____ * _____ )
            return _____
        ++s;
    }
    return * _____ == _____;
}

```

3 Dada las declaraciones `struct s {int m;} a[7]; int v, *f(int,int);`

Analice la siguiente expresión `((a[0]).m)=(v=(*f(1,((2+3)*4))))` y reescríbala removiendo todos los paréntesis innecesarios:

**Parte 2 - Afirmaciones**

- 1 Las GIC permiten definir la regla "toda variable debe declararse antes de usarse"... V F
- 2 El token constante entera 42 puede ser generado por diferentes lexemas..... V F
- 3 Para definir la GIC Sentencias es necesario definir la GIC Expresiones..... V F
- 4 El scanner agrega elementos a la tabla de símbolos ..... V F
- 5 El tipo da dato de una expresión C se determina en tiempo de compilación..... V F
- 6 Un PAS para Identificadores debe ser recursivo..... V F
- 7 La semántica estática es parte la fase de síntesis..... V F

Para el evaluador

Condiciones: ☐ 5 bien. ☐ Un ejercicio bien.

Bien(B) Mal(M) Sin hacer Total

Ejercicios(E)

Afirmaciones(A)

Nota: EB+AB-AM

Ejercicios(E)			
Afirmaciones(A)			
Nota: EB+AB-AM			

## Resolución

Una posible resolución del ejercicio 1

Falso. No recibe el tope de la pila, ni devuelve el string que debe apilarse.

Una posible solución, de muchas, es `int T(int e, int c, int t, char *s)`; donde el valor de retorno es el siguiente estado y en s se escribe la cadena que debe apilarse.

Resolución del ejercicio 2

```
int EsPalabraDeL3(const char *s){
    if(*s++ != 'a' )
        return 0;
    while(*s != '\0'){
        if( *s!='a'  && *s!='b' )
            return 0;
        ++s;
    }
    return *(s-1) == 'a' ;
}
```

Resolución del ejercicio 3

`a[0].m=v=*f(1, (2+3)*4)`

Afirmaciones

- 1 F
  - 2 V
  - 3 V
  - 4 V
  - 5 V
  - 6 F
  - 7 F
-