

UTN FRBA – SSL – Examen Final – 2020-12-21

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. En la especificación del Lenguaje de C está la siguiente gramática:

selection-statement:

`if (expression) statement`

`if (expression) statement else statement`

`switch (expression) statement`

- (1 punto) Indique el conjunto primero de *selection-statement*
 - (2 puntos) Justifique si la gramática es ambigua.
2. (2 puntos) Explique y ejemplifique en C el concepto de orden de evaluación de operandos.
3. Analice el siguiente fragmento C que prueba una función:

```
char user[1000];  
assert(0==strcmp("cjkent",CopyUser(user,"cjkent@dailyplanet.com")));
```

- (1 punto) Indique la cantidad de tokens en la *declaración* del fragmento anterior.
- (1 punto) Indique cuantos *literales* y *constantes* hay en la *sentencia* del fragmento anterior.
- (2 puntos) Escriba una expresión que use `user` y que sea semánticamente **incorrecta**. Justifique.
- (1 punto) Indique el prototipo de `CopyUser`.
- (2 puntos extra) Escriba la definición de `CopyUser`.

1. Una Resolución

1.
 - a. {if, switch}
 - b. Es ambigua porque hay más de una derivación posible para if con if-else anidado .
2. El orden de evaluación determina la secuencia de evaluación de los operandos de una expresión. C especifica el orden de evaluación de solo cuatro operadores. Cuando hay efecto de lado, el orden puede ser importante. Ejemplos: f(a, b) no se especifica si se evalúa primero f, a, ó b. a && b, sí, primero ese evalúa a y si es no cero, se evalúa b.
3.
 - a. Seis.
 - b. Tres: (0, "cj kent", "cj kent@dailyplanet.com")
 - c. Ejemplo: ++user. La expresión user no es un *lvalue* modificable.
 - d. `char* CopyUser(char*, const char*);`
 - e.

```
char* CopyUser(char* u, const char* a){
    char* p = u;
    while('@' != *a)
        *p++ = *a++;
    *p = '\0';
    return u;
}
```

v1.0.0-rc.3, 2021-02-07