

UTN FRBA – SSL – Examen Final – 2023-03-06

Apellido, Nombre:	Legajo:	Nota:
-------------------	---------	-------



- Resuelva el examen en tinta y en esta hoja; no se aceptan documentos adicionales.
- Durante el examen no se responden consultas; si lo necesita, escriba hipótesis de trabajo, las cuales también se evalúan.

1. (2 puntos) Justifique si la siguiente regex representa un sublenguaje de las constantes de C: $[3-7]^+L\{0,2\}U^?$
2. (2 puntos) En un proyecto con varios fuentes se tiene uno llamado `calculos.c` que incluye `calculos.h` y `funciones.h`. Escriba la regla de make para obtener su código objeto.

3. Dado el siguiente programa:

```
/* 1 */    #include<stdio.h>
/* 2 */    //Acá va la declaración de comprobar
/* 3 */    int main(void){
/* 4 */        float a = 3.2;
/* 5 */        //double a = 9.1;
/* 6 */        if (comprobar(a)) {
/* 7 */            //double a = 5.7;
/* 8 */            printf("dentro del if %g\n", a);
/* 9 */        }
/* 10 */        printf("fuera del if %g\n", a);
/* 11 */    }
/* 12 */    //Acá va la definición de comprobar
```

- a. (1 punto) Indique el tipo y el lexema del tercer token de la línea 6.
 - b. (1 punto) Dibuje el árbol de expresión para la expresión de la línea 6.
 - c. (1 punto) Declare `comprobar` para que su uso en 6 sea semánticamente correcto y permita compilar (generar código objeto, no ejecutable).
 - d. (1 punto) Si se descomenta solo la línea 5. ¿Compila correctamente? Justifique.
 - e. (1 punto) Si se descomenta solo la línea 7 ¿Compila correctamente? Justifique.
 - f. (1 punto) Defina `comprobar` para que permita construir el ejecutable e indique la salida de la ejecución.
4. (Punto extra) Declare `v` la siguiente expresión sea semánticamente correcta: `v[3]()`.

1. Una Resolución

1. Sí, porque son dígitos y al final agrega ninguna, una, o dos LL que es sufijo para variables de tipo long, y el U opcional lo hace unsigned long.

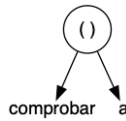
2.

```
calculos.o : calculos.c calculos.h funciones.h
cc -c calculos.c
```

3.

a. identificador comprobar

b.



c. bool comprobar(double); // int y float, entre otros, también válidos

d. No, error semántico por redifinición en el mismo alcance.

e. Sí, otro alcance para la nueva a.

f. bool comprobar(double d){return true;}

Salida:

dentro del if 3.2

fuera del if 3.2

4. void (*v[])(void);. La dimensión puede no definirse o ser mayor tres.

v1.0.0-beta.1 2023-03-05