

## Taller de Programación I

Fecha: 10 / 12 / 2018

Cuatrimestre: 2° / 2019

Tema: 1

Padrón: \_\_\_\_\_ Apellido: \_\_\_\_\_ Nombres: \_\_\_\_\_

Email: \_\_\_\_\_

Ejercicio	1	2	3	4	5	6	7	8	9	10
Resultado										

- 1) Explique breve y concretamente qué es **f**:  
`int (*f) (short int *, char[3]);`
- 2) Explique **qué es** cada uno de los siguientes, haciendo referencia a su **inicialización**, su **comportamiento** y el **area de memoria** donde residen:
  - a) Una variable **global static**
  - b) Una variable **local static**
  - c) Un **atributo de clase static**.
- 3) Escribir **un programa ISO C** que procese el archivo "nros2bytes.dat" **sobre sí mismo, duplicando los enteros de 2 bytes múltiplos de 3**.
- 4) ¿Cómo se logra que 2 **threads** accedan (lectura/escritura) a un mismo recurso compartido sin que se generen problemas de consistencia? **Ejemplifique**.
- 5) Escriba el **.H de una biblioteca** de funciones **ISO C** para cadenas de caracteres. Incluya, al menos, 4 funciones.
- 6) ¿Qué es una **macro** de **C**? Detalle las **buenas prácticas** para su definición. Ejemplifique
- 7) Describa **el proceso** de transformación de código fuente a un ejecutable. Precise las etapas y las tareas desarrolladas **en cada una de ellas**.
- 8) Indique la **salida** del siguiente programa:

```
class A      { A(){cout << "A()" << endl;}    ~A(){ cout << "~A()" << endl;} }
class B : public A { B(){cout << "B()" << endl;}    ~B(){ cout << "~B()" << endl;} }
int main () { B b; return 0;}
```
- 9) Implemente una función **C++** denominada **Sacar** que reciba dos listas de elementos y devuelva una nueva lista con los elementos de la primera que no están en la segunda:  
`std::list<T> Sacar(std::list<T> a, std::list<T> b);`
- 10) Escriba un programa que reciba por **línea de comandos** un **Puerto** y una **IP**. El programa debe establecer una unica conexión, quedar en escucha e **imprimir en stdout todo** lo recibido. Al recibir el texto 'FINAL' **debe finalizar** el programa **sin imprimir dicho texto**.