

# Guía de Trabajos Prácticos

Esta guía contiene los Trabajos Prácticos (TP) obligatorios del curso R1042, incluyendo una guía para su entrega con el formato de presentación y un cronograma con las fechas de entrega.

Durante el transcurso del año se deberán entregarán Trabajos Prácticos, que se caracterizan por ser incrementales. Esto significa que en cada TP se irán introduciendo nuevos conocimientos con respecto al TP anterior. A su vez, se deberán utilizar las herramientas SVN, Makefile y Doxygen cuando se lo indique.

## Formato de presentación

- Los archivos fuentes deben tener en todos los casos los comentarios necesarios para clarificar su lectura.
- Cada subrutina/función, debe contar con un encabezado describiendo la operación que realiza, los parámetros que espera como entrada, y los resultados que debe presentar, indicando formato y método de entrega.
- Como encabezado del programa, debe haber un comentario que explique claramente que hace dicho programa, y las instrucciones detalladas (comandos) para su compilación y “linkeo”.
- TODOS los ejercicios son obligatorios.
- **La entrega de TODOS los trabajos prácticos es obligatoria para regularizar la materia.**

# Trabajo Práctico N°5

## - Introducción a Punteros -

### Ejercicio 1

Responda las siguientes preguntas, use el tag `\note` de doxygen para colocar sus respuestas.

- \* ¿Qué función cumple el operador `&`?
- \* ¿Por qué cada vez que ejecuta el programa tienen direcciones distintas?
- \* ¿De qué depende la cantidad de bytes que ocupa la dirección de la variable?
- \* ¿Hay alguna relación entre el tipo de dato de una variable y el tamaño de su dirección?

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
int varIntA, varIntB;
```

```
char varCharA, varCharB;
```

```
printf ("varIntA = %08x; %p\r\n", varIntA, &varIntA);
```

```
printf ("varIntB = %08x; %p\r\n", varIntB, &varIntB);
```

```
printf ("varCharA = %08x; %p\r\n", varCharA, &varCharA);
```

```
printf ("varCharB = %08x; %p\r\n", varCharB, &varCharB);
```

```
return (0);
```

```
}
```

Nota: Revise: <https://www.stack.nl/~dimitri/doxygen/manual/commands.html#cmdnote>  
(referencia de tag `\note`)

## **Ejercicio 2**

Implemente un programa que utilizando un puntero de tipo unsigned char imprima byte a byte una variable de tipo int en hexadecimal que ha sido inicializada con el número 0x12345678. ¿El byte menos significativo del int en qué dirección se encuentra respecto al byte más significativo?

## **Ejercicio 3**

Implemente una función que realice las cuatro operaciones básicas entre dos números de tipo float y retorne el resultado. EL prototipo de la función es el siguiente:

float calculo(float opA, float opB, int op)

Donde:

- opA y opB son números con los cuales se debe realizar la operación
- op: La operación a realizar:
  1. Realiza la suma.
  2. Realiza la resta.
  3. Realiza el producto.
  4. Realiza la división.
- La función retorna el resultado NAN en caso de que ingrese una operación no válida.

*Nota: Para la implementación utilice puntero a función y estructura de selección switch.*

#### **Ejercicio 4**

Dado la siguiente declaración de un array:

```
int v[5] = {32,12,15,89,6};
```

y asumiendo que el sistema operativo nos asigna la dirección 0xbfff0000 para el comienzo del mismo, utilizando \note en doxygen responda:

- a) ¿Qué dirección de memoria contendrá el valor de v[2]?
- b) ¿Qué dirección de memoria resultará de resolver v+3?
- c) Indique cómo accedería al valor 89 dentro del vector, por cualquier método que conozca.
- d) Podemos recorrer el vector utilizando post-incremento (v++)? Justifique su respuesta.
- e) ¿Recibiríamos algún error durante la compilación el linkeo si tratáramos de acceder al contenido de v+10? En caso de que sí se pide justificar la respuesta. (No se admiten respuestas del tipo: “La consola imprime violación de segmento”)