

# Taller de Programación C++

**Ejercicio 1.** *Repetición controlada por un contador.* Escriba un programa que cargue y analice los resultados de los exámenes de un curso de 10 estudiantes. Las clasificaciones son números enteros en el rango  $[0, 10]$  y deben ser cargadas por teclado una a una, utilizando la función `cin`, vista el laboratorio 01. Muestre por pantalla la suma total de las calificaciones de todos los estudiantes y el promedio de la clase para este examen (con decimales).

**Ejercicio 2.** *Repetición controlada por un centinela.* Modificar el programa anterior para que el número de estudiantes sea arbitrario. Por ejemplo: **Ingrese la nota (-1 para terminar):**

**Ejercicio 3.** *Repetición controlada por un centinela y uso de contadores.* Modificar el ejercicio anterior de manera que cuente la cantidad de alumnos que aprobaron y reprobaron el examen (nota mayor a 6). La cantidad de alumnos total no puede ser menor a 6 (frente a este caso, mostrar un error por pantalla). En el caso de más de dos tercios de los alumnos hayan aprobado, indicar por pantalla un mensaje que avise que se puede incrementar el número de plazas en el curso.

**Ejercicio 4.** *Repetición controlada por un centinela y uso de acumuladores.* Escriba un programa que reciba como entrada un entero de tipo binario, conteniendo sólo 0s y 1s, para imprimir luego por pantalla el equivalente decimal. Se supone que la entrada es válida respetando la premisa de los 0s y 1s. En forma parecida al sistema numérico decimal, donde el dígito más a la derecha tiene un valor posicional de 1 y el siguiente dígito a la izquierda tiene un valor posicional de 10, después 100, después 1000, etc., en el sistema numérico binario, el dígito más a la derecha tiene un valor posicional de 1, el siguiente dígito a la izquierda tiene un valor posicional de 2, luego 4, luego 8, etc. Así, el número decimal 432 se puede interpretar como  $4 * 100 + 3 * 10 + 2 * 1$ . El equivalente decimal del número binario 1010 es  $0 * 1 + 1 * 2 + 0 * 4 + 1 * 8$ , o  $0 + 2 + 0 + 8$ , o 10. Ayuda: use los operadores módulo y división para elegir los dígitos del número binario uno a la vez, de derecha a izquierda.

**Ejercicio 5.** *Pasaje por valor y referencia.* Escriba y utilice en un programa una función que reciba el radio de un círculo (como un valor float), calcule e imprima el diámetro, la circunferencia y el área. Use el valor 3.14159 para  $\pi$ . Estos valores deben ser mostrados en pantalla desde el programa principal (main.cpp).

**Ejercicio 6.** *Pasaje por valor y referencia.* Escriba un programa completo en C++ con las dos funciones alternativas que se especifican a continuación, de las cuales cada una simplemente triplica la variable *cuenta* definida en main. Después compare y contraste ambos métodos. Estas dos funciones son:

1. la función `triplicarPorValor`, que pasa una copia de *cuenta* por valor, triplica la copia y devuelve el nuevo valor,
2. la función `triplicarPorReferencia`, que pasa *cuenta* por referencia a través de un parámetro por referencia y triplica el valor original de *cuenta* a través de su alias (es decir, el parámetro por referencia).

**Ejercicio 7.** Implementar la función `swap`: `void swap(int& a, int& b)`, que cumpla con la siguiente especificación:

```
proc swap (inout a:Z, inout b:Z) {
    Pre {a = a0 ∧ b = b0}
    Post {a = b0 ∧ b = a0}
}
```

**Ejercicio 8.** La conjetura de *Collatz* dice que dado un número natural  $n$  y el proceso que describimos a continuación, sin importar cuál sea el número original, provocará que la serie siempre termine en 1. El proceso:

- Si  $n$  es par lo dividimos por 2
- Si  $n$  es impar lo multiplicamos por 3 y le sumamos 1 al resultado

En este ejercicio, supondremos que la conjetura es cierta y se pide implementar una función que devuelva la cantidad de pasos que se realizan desde el número original hasta llegar a 1 y que muestre en la salida estándar la sucesión de números por la que pasa. Ejemplo: si calculamos collatz de 11, la cantidad de pasos es 15 y la sucesión es 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Resolver este ejercicio utilizando el siguiente encabezado para la función.

```
void collatz(int n, int& cantPasos)
```

Implementar en versiones iterativa y recursiva.

**Ejercicio 9.** Implementar la función `division` que cumpla con la siguiente especificación:

```
proc division (in dividendo  $\mathbb{Z}$ , in divisor  $\mathbb{Z}$ , out cociente: $\mathbb{Z}$ , out resto: $\mathbb{Z}$ ) {  
    Pre { $dividendo \geq 0 \wedge divisor > 0$ }  
    Post { $dividendo = divisor * cociente + resto \wedge 0 \leq resto < divisor$ }  
}
```

Resolver este ejercicio en versiones iterativa y recursiva.