

## Tarea 05:

Fecha de entrega: **ANTES** del 17 de Septiembre.

a) SE VALE VOLVERLO A ENVIAR (AYUDANTES, IGNOREN EL ANTERIOR Y EVALÚEN ESTE POR FAVOR) Hacer un programa que muestre "pruebe" la precedencia de los siguientes operadores (es decir ustedes tienen que proponer operaciones aritméticas y lógicas [verdadero, falso] mostrando que lo que sale a pantalla tienen sentido dado la forma en que la computadora primero opera algunos operadores y luego otros): `+`, `-`, `*`, `/`, `<=`, `>`, `&&`, `||` y `+=`. Ver <https://learn.microsoft.com/es-es/cpp/c-language/precedence-and-order-of-evaluation?view=msvc-170>

b) Hacer una función que reciba un valor `unsigned int v` por parámetro y que lo multiplique por 32 usando corrimientos a la izquierda. Reportar: ¿qué pasa si el resultado ya no cabe en un `unsigned int` después de los corrimientos?

c) Hacer un programa que adivine un número entre un valor inicial (por ejemplo 1) y un valor final (por ejemplo 1,000,000) en a los más  $\text{ceil}(\log_2(\text{final}-\text{inicial}+1+1))$  intentos (siendo `ceil()` la función *techo*). El programa pregunta las 3 opciones : ¿adivinado?, o es ¿mayor? o es ¿menor? ('a', '<', '>'). Deben de usar un ciclo `do-while`. Para hacerlo, programar una función que recibe el intervalo, haga la búsqueda, y regrese por la izquierda el número de intentos (iteraciones) que usó. `int main(void) { ... }` manda a llamar a dicha función e imprime cuantos intentos se utilizaron. Basen su programa en el código `lee_char_teclado.c`.

d) Ejecutar, comprender y **modificar** el programa que pone una animación de un conjunto de relojes en la pantalla, los archivos fuente se dan a continuación: `main_relojes.c` `screen.c` `screen.h`. Reportar la modificación que hicieron.

e) Hacer una animación interesante/bonita usando la función `gotoxy()` y ciclos `for` o `while`. Quizá pueden tomar ideas del *ASCII Art* (por

ejemplo de diferentes animales en <https://www.asciart.eu/animals/camels>) ¡sorprendan a su profesor! (gratamente).

f) Usar `#define` para indicar con máscaras de bits (que usan c/u un solo bit) que las siguientes configuraciones de una casa son verdaderas.

```
foco_prendido
auto_en_cochera
alarma_activada
puerta_cerrada
perro_suelto
ventana_cerrada
tinaco_de_agua_lleno
hay_luz_electrica
refrigerador_conectado
```

De tal forma que para generar un estado de la casa deben usar operaciones a bits (`&` , `|` , `!` ) en una sola variable `unsigned int` y que describa por completo el estado de una casa usando todas las configuraciones anteriores en VERDADERO o FALSO.

Hagan una función que recibe un único `unsigned int` y adentro imprime TODOS los mensajes correctos correspondientes a la configuración de una casa:

*El foco esta prendido o bien, El foco está apagado,  
La puerta está cerrada, o bien, La puerta está abierta,  
etc.*