

Almacenar en local y en repositorio remoto github modo privado con el usuario @jasdalinux o jasda@ingenieria.usac.edu.gt .
Siempre utilizar formato de reporte IEEE en un único documento pdf.

Lenguaje c

tarea08a.c (mejorado)

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

int main() {
    float *pesos, temp;
    int i, j, nest;
    system("clear");
    // Solicitar al usuario la cantidad de estudiantes
    printf("¿Cuántos estudiantes son?: ");
    if (scanf("%d", &nest) != 1 || nest <= 0) {
        // Manejar el caso de entrada inválida
        printf("Entrada inválida. Debe ingresar un número entero positivo.\n");
        return 1; // Salir del programa con código de error
    }

    // Reservar memoria dinámica para almacenar los pesos de los estudiantes
    pesos = (float *)malloc(nest * sizeof(float));
    if (pesos == NULL) {
        printf("Insuficiente Espacio de Memoria\n");
        return 1; // Salir del programa con código de error
    }

    // Ingresar los pesos de los estudiantes
    for (i = 0; i < nest; i++) {
        printf("Peso del Estudiante [%d]: ", i + 1);
        if (scanf("%f", (pesos + i)) != 1) {
            // Manejar el caso de entrada inválida
            printf("Entrada inválida. Debe ingresar un número real.\n");
            free(pesos); // Liberar la memoria asignada dinámicamente
            return 1; // Salir del programa con código de error
        }
    }

    printf("\n****ARRAY ORIGINAL ****\n");
    for (i = 0; i < nest; i++)
        printf("Peso[%d]: %.1f \n", i + 1, *(pesos + i));

    // Ordenar los pesos en orden ascendente utilizando el algoritmo de burbuja
    for (i = 0; i < nest; i++)
        for (j = 0; j < (nest - 1); j++)
            if (pesos[j] > pesos[j + 1]) {
                temp = pesos[j];
                pesos[j] = pesos[j + 1];
                pesos[j + 1] = temp;
            }

    printf("\n****ARRAY ORDENADO EN FORMATO ASCENDENTE****\n");
    for (i = 0; i < nest; i++)
        printf("Peso[%d]: %.1f \n", i + 1, *(pesos + i));

    // Liberar la memoria asignada dinámicamente
    free(pesos);

    return 0;
}
```

tarea08b.c

```
#include <gtk/gtk.h>
//gcc `pkg-config --cflags gtk+-3.0` -o programa8b programa8b.c `pkg-config --libs gtk+-3.0`

// Función para imprimir "Hello World"
static void print_hello(GtkWidget *widget, gpointer data) {
    g_print("Usando liberia GTK\n");
}

// Función para limpiar la consola
static void clear_console(GtkWidget *widget, gpointer data) {
    // Limpiar la consola (puedes adaptar esto dependiendo de tu sistema)
    system("clear");
}

// Función para salir del programa
static void quit_program(GtkWidget *widget, gpointer data) {
    g_print("Saliendo del programa\n"); // Salir del bucle de eventos de GTK
}

static void activate(GtkApplication *app, gpointer user_data) {
    GtkWidget *window;
    GtkWidget *button;
    GtkWidget *buttonBox;
```

```

window = gtk_application_window_new(app);
gtk_window_set_title(GTK_WINDOW(window), "Lenguaje C entorno grafico");
gtk_window_set_default_size(GTK_WINDOW(window), 400, 400);

buttonBox = gtk_button_box_new(GTK_ORIENTATION_VERTICAL);
gtk_container_add(GTK_CONTAINER(window), buttonBox);

button = gtk_button_new_with_label("Imprimir texto y salir");
g_signal_connect(button, "clicked", G_CALLBACK(print_hello), NULL);
g_signal_connect_swapped(button, "clicked", G_CALLBACK(gtk_widget_destroy), window);
gtk_container_add(GTK_CONTAINER(buttonBox), button);

// Botón para limpiar la consola
button = gtk_button_new_with_label("Limpiar");
g_signal_connect(button, "clicked", G_CALLBACK(clear_console), NULL);
gtk_container_add(GTK_CONTAINER(buttonBox), button);

gtk_widget_show_all(window);
}

int main(int argc, char **argv) {
    GtkApplication *app;
    int status;

    app = gtk_application_new("org.gtk.example", G_APPLICATION_FLAGS_NONE);
    g_signal_connect(app, "activate", G_CALLBACK(activate), NULL);
    status = g_application_run(G_APPLICATION(app), argc, argv);
    g_object_unref(app);

    return status;
}

```

<https://www.guru99.com/es/c-programming-tutorial.html>
<https://github.com/jasdalinux/jasdalinux2024>