

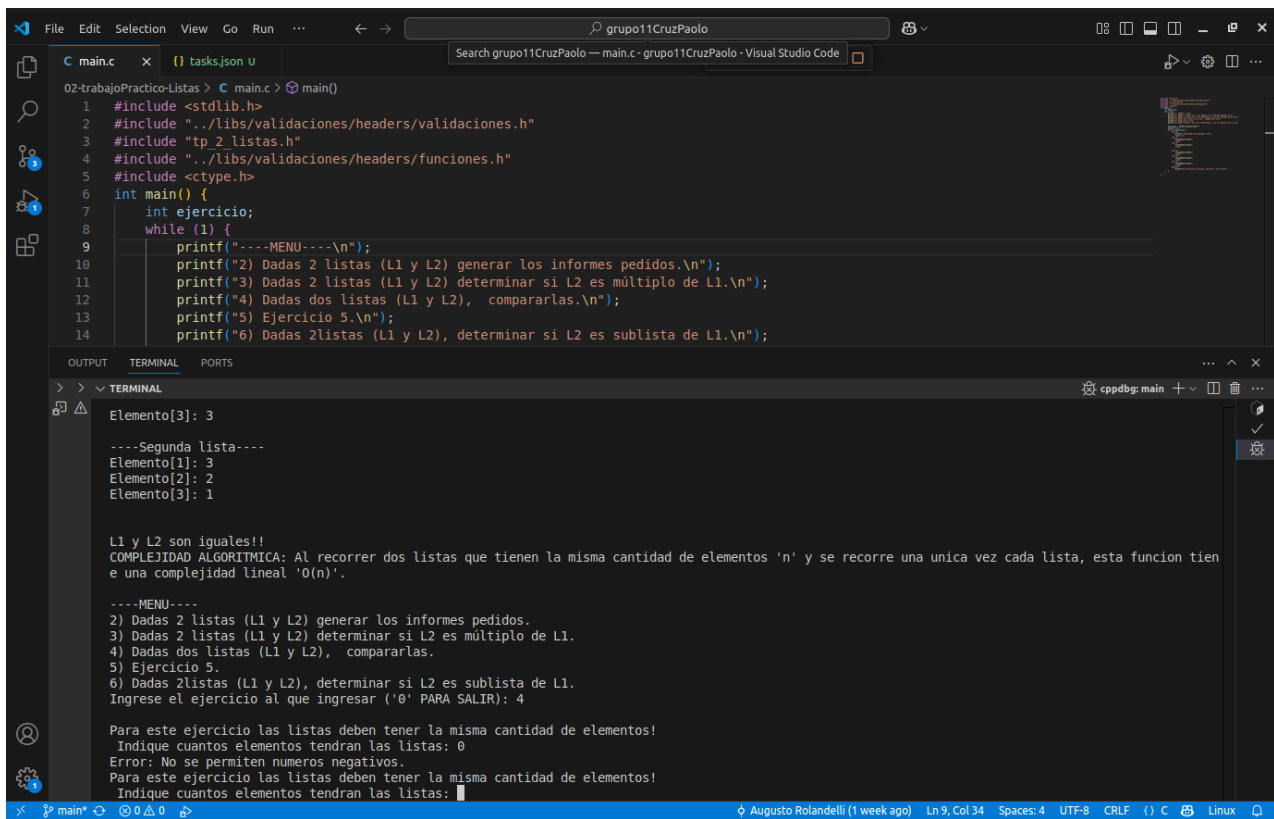
RESULTADO DE LA CORRECCIÓN: **DESAPROBADO**

## OBSERVACIONES GENERALES A TODOS LOS GRUPOS

En caso de no haber resultado para mostrar, por ejemplo no se encontró coincidencias en lo buscado, indicarlo por pantalla. Si no se muestra nada no queda claro el resultado.

## OBSERVACIONES

No permite listas vacías.



The screenshot shows the Visual Studio Code editor with a C program named `main.c` and its terminal output. The program is a menu-driven application for linked lists. The terminal output shows the execution of the program, including the menu, user input, and the program's response.

```
02-trabajoPractico-Listas > C main.c > main()
1 #include <stdlib.h>
2 #include "../libs/validaciones/headers/validaciones.h"
3 #include "tp_2_listas.h"
4 #include "../libs/validaciones/headers/funciones.h"
5 #include <ctype.h>
6 int main() {
7     int ejercicio;
8     while (1) {
9         printf("----MENU----\n");
10        printf("2) Dadas 2 listas (L1 y L2) generar los informes pedidos.\n");
11        printf("3) Dadas 2 listas (L1 y L2) determinar si L2 es múltiplo de L1.\n");
12        printf("4) Dadas dos listas (L1 y L2), compararlas.\n");
13        printf("5) Ejercicio 5.\n");
14        printf("6) Dadas 2listas (L1 y L2), determinar si L2 es sublista de L1.\n");
15    }
16}
```

Terminal Output:

```
Elemento[3]: 3
----Segunda lista----
Elemento[1]: 3
Elemento[2]: 2
Elemento[3]: 1

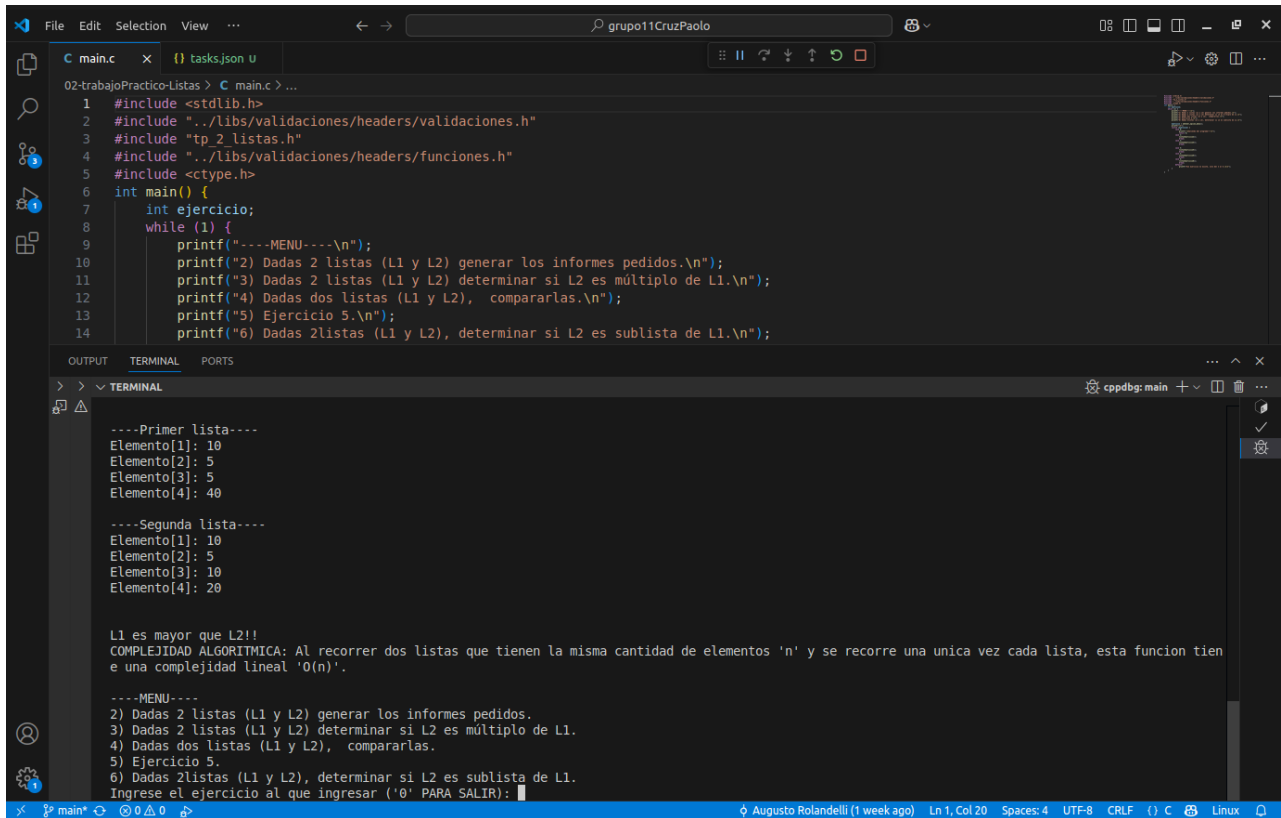
L1 y L2 son iguales!!
COMPLEJIDAD ALGORITMICA: Al recorrer dos listas que tienen la misma cantidad de elementos 'n' y se recorre una unica vez cada lista, esta funcion tien e una complejidad lineal 'O(n)'.

----MENU----
2) Dadas 2 listas (L1 y L2) generar los informes pedidos.
3) Dadas 2 listas (L1 y L2) determinar si L2 es múltiplo de L1.
4) Dadas dos listas (L1 y L2), compararlas.
5) Ejercicio 5.
6) Dadas 2listas (L1 y L2), determinar si L2 es sublista de L1.
Ingrese el ejercicio al que ingresar ('0' PARA SALIR): 4

Para este ejercicio las listas deben tener la misma cantidad de elementos!
Indique cuantos elementos tendran las listas: 0
Error: No se permiten numeros negativos.
Para este ejercicio las listas deben tener la misma cantidad de elementos!
Indique cuantos elementos tendran las listas: 
```

## GRUPO 11 – Correcciones Trabajo Práctico: LISTAS

Ejercicio 4: no siempre detecta correctamente si las listas son iguales.



```
1 #include <stdlib.h>
2 #include "../libs/validaciones/headers/validaciones.h"
3 #include "tp_2_listas.h"
4 #include "../libs/validaciones/headers/funciones.h"
5 #include <ctype.h>
6 int main() {
7     int ejercicio;
8     while (1) {
9         printf("----MENU----\n");
10        printf("2) Dadas 2 listas (L1 y L2) generar los informes pedidos.\n");
11        printf("3) Dadas 2 listas (L1 y L2) determinar si L2 es múltiplo de L1.\n");
12        printf("4) Dadas dos listas (L1 y L2), compararlas.\n");
13        printf("5) Ejercicio 5.\n");
14        printf("6) Dadas 2listas (L1 y L2), determinar si L2 es sublista de L1.\n");
15    }
16    while (1) {
17        printf("----Primer lista----\n");
18        Elemento[1]: 10
19        Elemento[2]: 5
20        Elemento[3]: 5
21        Elemento[4]: 40
22
23        printf("----Segunda lista----\n");
24        Elemento[1]: 10
25        Elemento[2]: 5
26        Elemento[3]: 10
27        Elemento[4]: 20
28
29        L1 es mayor que L2!!
30        COMPLEJIDAD ALGORITMICA: Al recorrer dos listas que tienen la misma cantidad de elementos 'n' y se recorre una unica vez cada lista, esta funcion tien e una complejidad lineal 'O(n)'.
31
32        printf("----MENU----\n");
33        printf("2) Dadas 2 listas (L1 y L2) generar los informes pedidos.\n");
34        printf("3) Dadas 2 listas (L1 y L2) determinar si L2 es múltiplo de L1.\n");
35        printf("4) Dadas dos listas (L1 y L2), compararlas.\n");
36        printf("5) Ejercicio 5.\n");
37        printf("6) Dadas 2listas (L1 y L2), determinar si L2 es sublista de L1.\n");
38    }
39    printf("Ingrese el ejercicio al que ingresar ('0' PARA SALIR): ");
40}
```

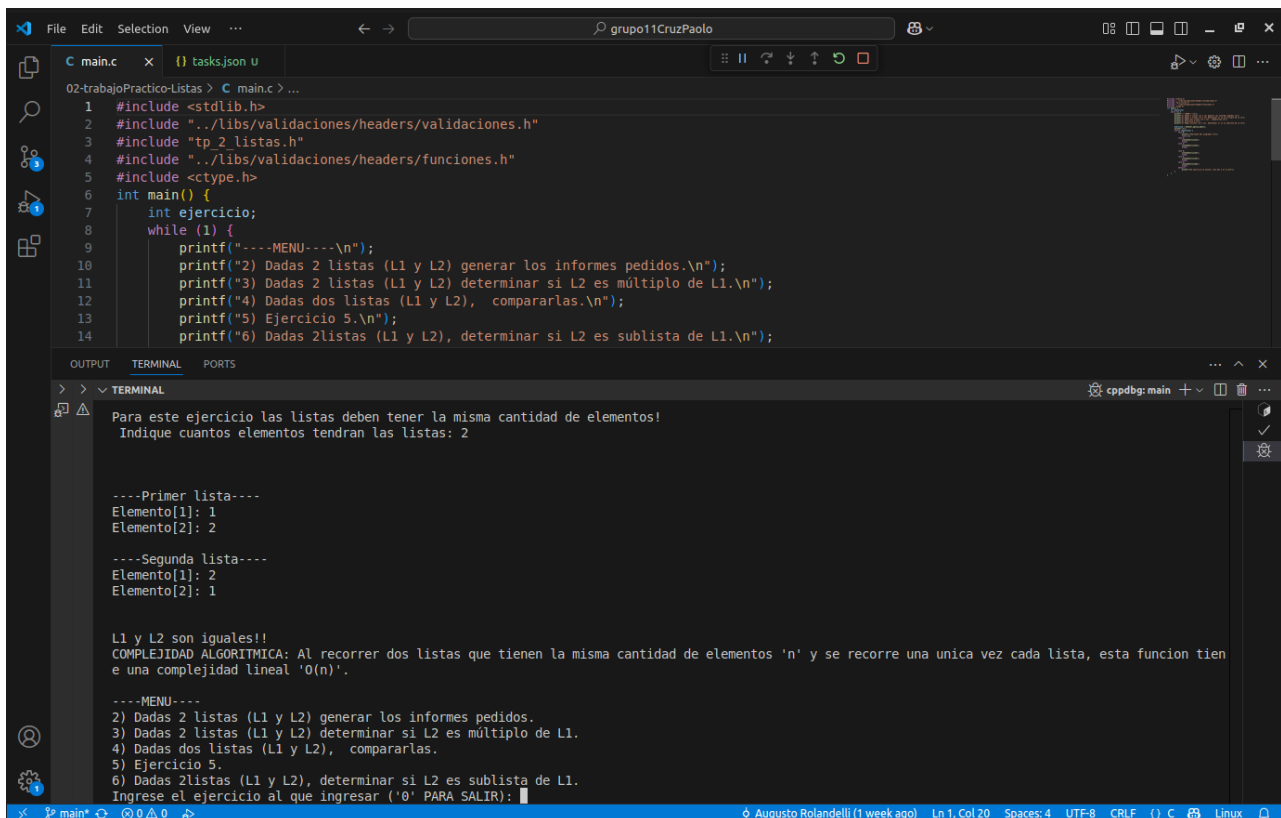
Terminal Output:

```
----Primer lista----
Elemento[1]: 10
Elemento[2]: 5
Elemento[3]: 5
Elemento[4]: 40

----Segunda lista----
Elemento[1]: 10
Elemento[2]: 5
Elemento[3]: 10
Elemento[4]: 20

L1 es mayor que L2!!
COMPLEJIDAD ALGORITMICA: Al recorrer dos listas que tienen la misma cantidad de elementos 'n' y se recorre una unica vez cada lista, esta funcion tien e una complejidad lineal 'O(n)'.

----MENU----
2) Dadas 2 listas (L1 y L2) generar los informes pedidos.
3) Dadas 2 listas (L1 y L2) determinar si L2 es múltiplo de L1.
4) Dadas dos listas (L1 y L2), compararlas.
5) Ejercicio 5.
6) Dadas 2listas (L1 y L2), determinar si L2 es sublista de L1.
Ingrese el ejercicio al que ingresar ('0' PARA SALIR):
```



```
1 #include <stdlib.h>
2 #include "../libs/validaciones/headers/validaciones.h"
3 #include "tp_2_listas.h"
4 #include "../libs/validaciones/headers/funciones.h"
5 #include <ctype.h>
6 int main() {
7     int ejercicio;
8     while (1) {
9         printf("----MENU----\n");
10        printf("2) Dadas 2 listas (L1 y L2) generar los informes pedidos.\n");
11        printf("3) Dadas 2 listas (L1 y L2) determinar si L2 es múltiplo de L1.\n");
12        printf("4) Dadas dos listas (L1 y L2), compararlas.\n");
13        printf("5) Ejercicio 5.\n");
14        printf("6) Dadas 2listas (L1 y L2), determinar si L2 es sublista de L1.\n");
15    }
16    while (1) {
17        printf("----Primer lista----\n");
18        Elemento[1]: 1
19        Elemento[2]: 2
20
21        printf("----Segunda lista----\n");
22        Elemento[1]: 2
23        Elemento[2]: 1
24
25        L1 y L2 son iguales!!
26        COMPLEJIDAD ALGORITMICA: Al recorrer dos listas que tienen la misma cantidad de elementos 'n' y se recorre una unica vez cada lista, esta funcion tien e una complejidad lineal 'O(n)'.
27
28        printf("----MENU----\n");
29        printf("2) Dadas 2 listas (L1 y L2) generar los informes pedidos.\n");
30        printf("3) Dadas 2 listas (L1 y L2) determinar si L2 es múltiplo de L1.\n");
31        printf("4) Dadas dos listas (L1 y L2), compararlas.\n");
32        printf("5) Ejercicio 5.\n");
33        printf("6) Dadas 2listas (L1 y L2), determinar si L2 es sublista de L1.\n");
34    }
35    printf("Ingrese el ejercicio al que ingresar ('0' PARA SALIR): ");
36}
```

Terminal Output:

```
Para este ejercicio las listas deben tener la misma cantidad de elementos!
Indique cuantos elementos tendran las listas: 2

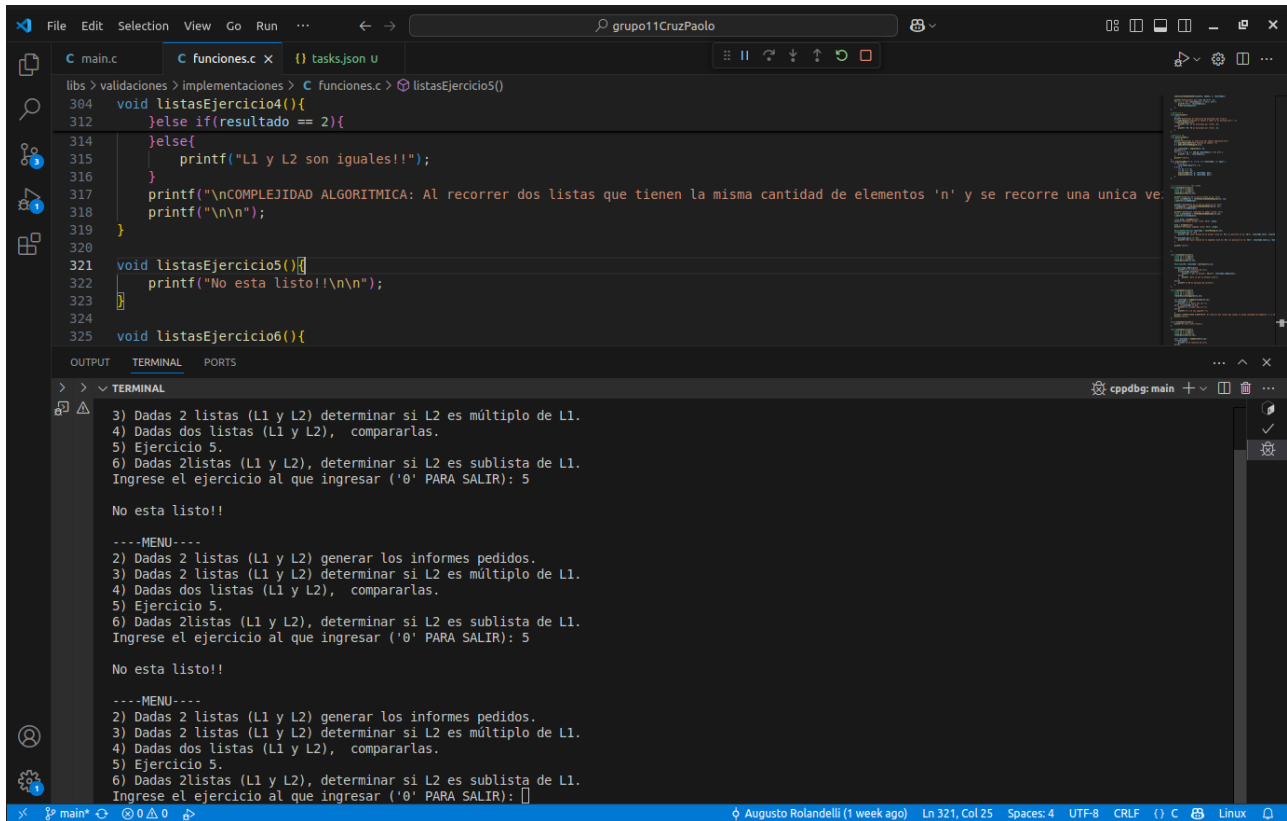
----Primer lista----
Elemento[1]: 1
Elemento[2]: 2

----Segunda lista----
Elemento[1]: 2
Elemento[2]: 1

L1 y L2 son iguales!!
COMPLEJIDAD ALGORITMICA: Al recorrer dos listas que tienen la misma cantidad de elementos 'n' y se recorre una unica vez cada lista, esta funcion tien e una complejidad lineal 'O(n)'.

----MENU----
2) Dadas 2 listas (L1 y L2) generar los informes pedidos.
3) Dadas 2 listas (L1 y L2) determinar si L2 es múltiplo de L1.
4) Dadas dos listas (L1 y L2), compararlas.
5) Ejercicio 5.
6) Dadas 2listas (L1 y L2), determinar si L2 es sublista de L1.
Ingrese el ejercicio al que ingresar ('0' PARA SALIR):
```

Ejercicio 5: no lo resolvieron.



Ejercicio 6: compara en un solo sentido. Complejidad incorrecta.