

## GRUPO 1 – Correcciones Trabajo Práctico: RECURSIVIDAD

**RESULTADO DE LA CORRECCIÓN: DESAPROBADO**

## OBSERVACIONES

Crear un único archivo .c en donde se realicen las implementaciones de lo definido en el .h para facilitar los tests automáticos. Para funciones propias pueden crear sus propios .h y .c o sumarlas al .h subido a la plataforma. El llamado de las funciones realizado en main.c es correcto.

En algunos ejercicios, ingresé una letra y entra en un loop, en otros casos me saca del programa directamente.

Utiliza la sentencia for para resoluciones relacionadas con la recursividad.

El ejercicio 10 no funciona con todos los valores probados.

## GRUPO 1 – Correcciones Trabajo Práctico: RECURSIVIDAD

The screenshot shows the Visual Studio Code editor with a C program. The main function calls 'continuar(10)'. The 'continuar' function is defined as follows:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <assert.h>
4 #include <stdbool.h>
5 #include <ctype.h>
6 #include <string.h>
7 #include "TPI_Recursividad.h"
8
9 void continuar(int *flag)
10 {
11     char *check = (char *)malloc(100 * sizeof(char));
12
13     printf("\nDesea continuar? (Si/No): ");
14     scanf("%99s", check);
15     // Convertir la cadena a mayusculas
16     for (int i = 0; check[i] != '\0'; i++)
```

An exception has occurred, and the program is aborted. The terminal output shows the user inputting '20' for the number and '5' for the bomb, followed by the error message 'malloc(): corrupted top size'.

Variables:   
check: 0x0  
flag: 0x7fffffffdb1c  
Registers:   
WATCH:   
CALL STACK:   
continuar(int \* flag) m...  
test10() main.c (225:1)  
main() main.c (280:1)

The screenshot shows the Visual Studio Code editor with a C program. The main function calls 'test10()'. The 'test10' function is defined as follows:

```
203 void test10()
204 {
205     int numero;
206     while (flag == 0)
207     {
208         int n;
209         int b;
210         printf("\n Ingrese el numero: ");
211         scanf("%i", &n);
212         printf("\n Ingrese la boma: ");
213         scanf("%i", &b);
214
215         int max = n / b + 1;
216         int *resultado = malloc((max + 1) * sizeof(int));
217
218         resultado = explosion(n, b);
219         for (int i = 0; i <= max; i++)
220         {
221             // ...
222         }
223     }
224 }
```

An exception has occurred, and the program is aborted. The terminal output shows the user inputting '10' for the number, '3' for the bomb, and 'Si' for the continuation prompt. The error message 'malloc(): corrupted top size' is displayed.

Variables:   
n: 30  
b: 5  
max: 7  
resultado: 0x555555559b70  
flag: 0  
Registers:   
WATCH:   
CALL STACK:   
test10() main.c (218:1)  
main() main.c (280:1)