	<p style="text-align: center;">UNIVERSIDAD DE COSTA RICA ESCUELA DE INGENIERÍA ELÉCTRICA</p> <p style="text-align: center;">PROGRAMA DEL CURSO VIRTUAL</p>	<p style="text-align: center;">EIE</p> <hr/> <p style="text-align: center;">Escuela de Ingeniería Eléctrica</p>
<p style="text-align: center;">IE-0117 Programación Bajo Plataformas Abiertas I-2025</p>		

Laboratorio #4

- Todos los laboratorios deben venir con su respectivo reporte escrito en \LaTeX , el cual tendrá un valor de 25 pts. Sin embargo, si no presenta reporte, el laboratorio no se revisará.
- El reporte debe contener una explicación de su implementación, resultados y una discusión de los mismos. Asegúrese además de mostrar en los resultados los diferentes casos que solicita el enunciado.
- Incluya en el reporte una sección de **Repositorio**, en el cual agrega el link al repositorio de GitHub donde tiene el código.
- Puede incluir snippets del código en el reporte si así lo desea, pero esto **no contará** como discusión de resultados.
- Debe incluir capturas de pantalla para justificar sus resultados, limitándolas a sólo la información relevante. No agregue screenshots de toda la pantalla.
- Cuide siempre el formato y la legibilidad de su reporte. Sea conciso y presente sólo **información relevante**.
- Asegúrese de que su código compila. Si un ejercicio no compila, no será revisado.
- No se tolerará el plagio.
- Implemente siempre las mejores prácticas vistas en clase. Asegúrese de que su código está bien indentado y documentado.
- Todo el código deberá ser entregado a través de GitHub, de lo contrario no será revisado. Para ello, debe generar un repositorio público e incluir su enlace en el reporte en una sección destinada a esto.

Ejercicio 1

Considere el siguiente código como base, basado en el laboratorio anterior.

```

1 #include <stdio.h>
2
3 /*
```



UNIVERSIDAD DE COSTA RICA
ESCUELA DE INGENIERÍA ELÉCTRICA

PROGRAMA DEL CURSO VIRTUAL

EIE

Escuela de
Ingeniería Eléctrica

IE-0117 Programación Bajo Plataformas Abiertas
I-2025

```
4  Funcion para encontrar la longitud de la linea mas larga de 1s en una matriz
   cuadrada de tamaño size.
5
6  Params
7  — int **matrix: Puntero a la matriz cuadrada de enteros.
8  — int size: Tamaño de la matriz (numero de filas y columnas).
9  — int *result: Puntero donde se almacenara la longitud de la secuencia mas
   larga de 1s.
10
11  Retorno:
12  — No retorna un valor, pero modifica el valor en result.
13  */
14 void findLargestLine(int **matrix, int size, int * result) {
15
16 }
17
18 /*
19 Funcion para reservar la matriz en memoria dinamica.
20
21 Params:
22 — int ***matrix: Doble puntero para la matriz que sera creada.
23 — int size: Tamaño de la matriz (numero de filas y columnas).
24
25 Retorno:
26 — No retorna un valor. Reserva memoria para la matriz.
27 */
28 void allocateMatrix(int ***matrix, int size) {
29 }
30
31 /*
32 Funcion para llenar la matriz con numeros aleatorios (0s y 1s).
33
34 Parametros:
35 — int **matrix: Puntero a la matriz a llenar.
36 — int size: tamaño de la matriz (numero de filas y columnas).
37
38 Retorno:
39 — No retorna un valor. Llena la matriz con valores aleatorios.
40 */
41 void fillMatrix(int **matrix, int size) {
42 }
43
44 /*
45 Funcion para imprimir la matriz.
```



UNIVERSIDAD DE COSTA RICA
ESCUELA DE INGENIERÍA ELÉCTRICA


PROGRAMA DEL CURSO VIRTUAL

EIE

Escuela de
Ingeniería Eléctrica

IE-0117 Programación Bajo Plataformas Abiertas
I-2025

```
46
47 Params:
48 - int **matrix: Puntero a la matriz que se va a imprimir.
49 - int size: tamaño de la matriz (numero de filas y columnas).
50
51 Retorno:
52 - No retorna un valor. Imprime la matriz en la salida estandar.
53 */
54 void printMatrix(int **matrix, int size) {
55     printf("Matrix (%dx%d):\n", size, size);
56     for (int i = 0; i < size; i++) {
57         for (int j = 0; j < size; j++) {
58             printf("%d ", (*(matrix + i) + j));
59         }
60         printf("\n");
61     }
62 }
63
64 /*
65 Funcion para liberar la memoria asignada a la matriz.
66
67 Params:
68 - int **matrix: Puntero a la matriz que se va a liberar.
69 - int size: tamaño de la matriz (numero de filas y columnas).
70
71 Retorno:
72 - No retorna un valor. Libera la memoria utilizada por la matriz.
73 */
74 void freeMatrix(int **matrix, int size) {
75     for (int i = 0; i < size; i++) {
76         free(matrix[i]);
77     }
78     free(matrix);
79 }
80
81 int main() {
82     int size, largestLine;
83     int **matrix = NULL;
84     ...
85
86     findLargestLine(matrix, size, &largestLine);
87     ...
88     printf("El tamaño de la secuencia de ls mas grande es: %d\n", largestLine)
89     ;
```

	<p style="text-align: center;">UNIVERSIDAD DE COSTA RICA ESCUELA DE INGENIERÍA ELÉCTRICA</p> <p style="text-align: center;">PROGRAMA DEL CURSO VIRTUAL</p>	<p>EIE</p> <hr/> <p>Escuela de Ingeniería Eléctrica</p>
<p style="text-align: center;">IE-0117 Programación Bajo Plataformas Abiertas I-2025</p>		

```

89
90     return 0;
91 }

```

1. Escriba un programa que recorra, **utilizando aritmética de punteros y no la sintaxis de indexado**, una matriz cuadrada binaria y determine la cantidad de 1s consecutivos más larga que encuentra, incluyendo los 1s que continúan en la siguiente fila (no es necesario que considere columnas o diagonales). La matriz tiene un número fijo de filas y columnas. El programa debe devolver la longitud de la secuencia más larga de 1s encontrados.
2. Como parte de discusión en su reporte, agregue el pseudocódigo de su diseño y la explicación del mismo.
3. Utilice la lógica del laboratorio anterior para llenar la matriz de manera aleatoria, considerando siempre que sea binaria y cuadrada. Modifique su código de manera tal que las dimensiones de la matriz generada sean ingresadas por el usuario. Para ello, deberá hacer uso de memoria dinámica.

Consideraciones adicionales:


- Asegúrese de ejemplificar diferentes dimensiones de matrices en su reporte.
- Adicionalmente, su código no deberá tener memory leaks. Para ello, deberá incluir en el reporte el resultado de ejecutar su programa con valgrind, demostrando que no tiene memory leaks.
- Su código deberá compilar sin errores ni warnings con el siguiente comando:

```
1 gcc -Wall ejercicio1.c -o ej1
```

- Ejemplifique cada caso en su laboratorio, con la matriz utilizada y la secuencia encontrada.

Ejercicio 2

Escriba un programa en C que lea palabras desde un archivo de texto (input.txt) y encuentre la palabra más larga que sea un palíndromo. Para ello, asegúrese de que su programa realiza lo siguiente:

	<p style="text-align: center;">UNIVERSIDAD DE COSTA RICA ESCUELA DE INGENIERÍA ELÉCTRICA</p> <p style="text-align: center;">PROGRAMA DEL CURSO VIRTUAL</p>	<p style="text-align: center;">EIE</p> <hr/> <p>Escuela de Ingeniería Eléctrica</p>
<p>IE-0117 Programación Bajo Plataformas Abiertas I-2025</p>		

1. Leer el archivo palabra por palabra, ignorando signos de puntuación.
2. Para cada palabra:
 - a) Convierta cada palabra a minúscula.
 - b) Elimine caracteres no alfanuméricos.
 - c) Verifique si es un palíndromo.

Al final, imprima el palíndromo mas largo encontrado. Consideraciones adicionales:

- Asegúrese de que su código no tiene memory leaks. Para ello, deberá incluir en el reporte el resultado de ejecutar su programa con valgrind, demostrando que no tiene memory leaks.
- Su código deberá compilar sin errores ni warnings con el siguiente comando:

```
1 gcc -Wall ejercicio2.c -o ej2
```

Desglose de evaluación:


Ejercicio 1 (50 pts)

- Funcionalidad: el programa realiza la funcionalidad solicitada, a través de punteros. Si lo realiza de manera indexada, perderá los puntos. (20 pts)
- Funcionalidad: utiliza memoria dinámica para crear una matriz de contenido generado aleatoriamente y cuyas dimensiones el usuario controla. (20pts)
- No memory leaks. (5 pts)
- Código bien indentado y documentado: (5 pts)

Ejercicio 2 (20 pts)

- Funcionalidad: el programa realiza la funcionalidad solicitada. (15 pts)
- No memory leaks. (2 pts)
- Código bien indentado y documentado: (3 pts)

Reporte (25 pts)

	<p>UNIVERSIDAD DE COSTA RICA ESCUELA DE INGENIERÍA ELÉCTRICA</p> <p>PROGRAMA DEL CURSO VIRTUAL</p>	<p>EIE Escuela de Ingeniería Eléctrica</p>
<p>IE-0117 Programación Bajo Plataformas Abiertas I-2025</p>		

- Formato: en latex o markdown, hace un uso correcto del formato. (4 pts)
- Contenido: apropiada discusión de resultados, con muestras de los resultados obtenidos por cada ejercicio. (21 pts)

Git (5pts)

- Nombró los archivos según se solicitó (2 pts)
- Creó un branch para el ejercicio y lo integró a main. (3pts)