

Licenciatura en Ingeniería de Software
“Programación Estructurada”
M.G.T.I. María Enriqueta Castellanos Bolaños, E.D.
Segundo Parcial

Fecha: 11 de junio de 2020

Nombre: _____ Calificación: _____

Nombre del archivo depositado en la plataforma: _____

Instrucciones generales:

- Lee cuidadosamente toda la prueba antes de iniciar.
- La prueba tiene un valor de 35 puntos y representa el 35% de la calificación final del curso.
- Dispones de 1 hora con 20 minutos para resolver la prueba.
- Durante el examen:
 - o No se resolverán dudas relacionadas sobre cómo codificar el programa.
 - o No podrás solicitar que se evalúe el procedimiento que hasta el momento hayas desarrollado.
 - o Deberás resolver por ti mismo los errores de compilación que se te presenten.
 - o No está permitido mantener comunicación con otros estudiantes.
 - o Está prohibido consultar fuentes impresas y/o electrónicas.
- Queda estrictamente prohibido el uso de teléfonos celulares.
- Asegúrate de nombrar el archivo de código fuente de acuerdo al siguiente formato: <nombre del alumno en minúsculas y sin espacios en blanco><apellido paterno en minúsculas y sin espacios en blanco>. Por ejemplo, para alguien cuyo nombre sea Juan Pérez Ruiz, el nombre de su archivo sería: juanperez.cpp.
- El código fuente deberá tener una indentación o sangría de 3 espacios en blanco.
- Una vez que hayas finalizado el examen, deberás depositar el archivo de código fuente (.cpp) en la sección **Exámenes parciales** del curso en línea.
- Se descontará un punto por cada minuto de retraso posterior a la hora de entrega señalada en la plataforma.
- Únicamente serán revisados aquellos programas que compilen. Se recomienda comentar las líneas de código que arrojen mensajes de error, con el fin de no perder la totalidad de los puntos.
- No se admitirán correcciones al programa una vez que hayas hecho entrega formal de éste en la plataforma.

I. Instrucciones: Analiza la situación que a continuación se plantea y desarrolla un programa que se ajuste a lo especificado en los incisos a, b, c, d y e:

Se desea implementar un programa en C que reciba como entrada una matriz cuadrada de enteros y calcule la transpuesta de esa matriz. La transpuesta de una matriz A viene dada por la matriz que se obtiene cambiando la fila i por la columna i. Es decir, A^T es la matriz transpuesta de A si se tiene que $A^T[i,j]=A[j,i]$.

Por ejemplo si tenemos la siguiente matriz A:

$$A = \begin{bmatrix} 5 & 6 & 4 \\ 2 & 7 & 9 \\ 1 & 3 & 8 \end{bmatrix}$$

La transpuesta de dicha matriz es la siguiente:

$$A^T = \begin{bmatrix} 5 & 2 & 1 \\ 6 & 7 & 3 \\ 4 & 9 & 8 \end{bmatrix}$$

Datos:

`MATRIZ[1...N, 1...N]` `1 <= N <= 10`

Donde:

`MATRIZ` es un arreglo bidimensional de enteros, cuya capacidad máxima es de 100 elementos.

`N` es una variable de tipo entero, que expresa el orden de la matriz. Su valor debe estar comprendido en el intervalo `[1,10]`.

Las tareas que el programa deberá realizar se describen con detalle en los siguientes incisos:

- a) Verificar que el orden máximo de la matriz sea 10, es decir, que no exceda de 10 filas por 10 columnas (`#define MAX 10`). Para ello, deberás crear una función que se ajuste al prototipo que se expone seguidamente e invocarla dentro de una estructura `do while` desde la función principal del programa. **(8 puntos)**:

```
bool validar(int orden);
```

- b) Leer una matriz cuadrada. Para ello, deberás crear una función que se ajuste al prototipo que se expone seguidamente **(2 puntos)**:

```
void leerArreglo(int arreglo[][MAX], int orden);
```

Nota: La captura de los elementos la matriz deberá apegarse al siguiente formato:

```
Matriz[0][0]:
Matriz[0][1]:
Matriz[0][2]:
...
Matriz[9][9]:
```

- c) Calcular la traspuesta de una matriz cuadrada **sobre si misma** (sin utilizar una matriz auxiliar). Para ello, deberás crear una función que se ajuste al prototipo que se expone seguidamente **(16 puntos)**:

```
void calcularTranspuesta(int arreglo[][MAX], int orden);
```

- d) Imprimir los elementos de la matriz transpuesta calculada, en formato tabular **(7 puntos)**:

```
void imprimirArreglo(int arreglo[][MAX], int orden);
```

Nota: La impresión de la matriz transpuesta deberá apegarse al siguiente formato:

```
5    2    1
6    7    3
4    9    8
```

- e) Indentar adecuadamente el código fuente **(2 puntos)**. La puntuación será asignada sólo si el código fuente compila.

Consideraciones importantes:

- La indentación del código fuente deberá ser de 3 espacios en blanco.
- El programa deberá ser implementado de acuerdo a lo establecido en los incisos anteriores. Si no cumple esos criterios, será imposible verificar el dominio de los conceptos evaluados.
- No serán otorgadas proporciones de la puntuación asignada a cada inciso, por implementaciones incorrectas, incompletas o mal estructuradas.
- Se descontarán puntos por incurrir en malas prácticas de programación.

Criterios de evaluación:

- Cumplimiento de los lineamientos establecidos en la prueba.
- Planteamiento claro y coherente de la solución a la problemática presentada.
- Dominio en la aplicación de los conceptos fundamentales de la metodología de la programación en la solución de problemas.
- Implementación correcta de la solución al problema planteado.
- Uso correcto de las instrucciones del lenguaje de programación de alto nivel.
- Uso de una herramienta de depuración de código fuente.
- Ausencia de errores sintácticos (o de compilación).
- Ausencia de errores semánticos.
- Desempeño exitoso de la ejecución del programa.
- *Uso de estándares de codificación.*
- Seguimiento de buenas prácticas de programación.