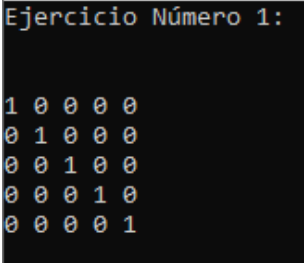


TP Anual Programación I

Indicaciones y Pautas de entrega (leer con atención):

- a- Deberán desarrollar/generar un algoritmo (un programa), por cada punto, que resuelva la problemática que se plantea. Los mismos deberán estar desarrollados en C#.
- b- La entrega de todos los desarrollos, en un único documento y con el formato requerido, **tiene como fecha límite el miércoles 30 de Octubre** (hasta las 23:59 y del año en curso) siendo como único medio de entrega el envío del mismo desde un mensaje o mail en la plataforma que más adelante se les indique (más detalles al final de este documento).
- c- El formato de entrega es en pdf o Word, indicando punto de la guía y pegando en el mismo dos cosas: el código del programa (en modo texto para poder copiar y pegar) y una captura de pantalla de su salida o resultado de ejecución (esto sí a modo imagen). A modo de ejemplo el siguiente desarrollo e imagen (recomiendo, y entiéndase como obligatorio, usar una tabla por cuestiones de prolijidad tal y como se muestra en el ejemplo):

Ejercicio 1	
<pre>using System; namespace Ejercicio1 { class Program { public static void Main(string[] args) { int[,] mat = new int[5, 5]; Console.WriteLine("Ejercicio Número 1:"); Console.WriteLine("\n"); for (int f = 0; f < 5; f++) { for (int c = 0; c < 5; c++) { mat[f, c] = 0; if (f == c) { mat[f, c] = 1; } Console.Write(mat[f, c] + " "); } Console.WriteLine(" "); } Console.ReadLine(); } } }</pre>	 <p>Ejercicio Número 1:</p> <pre>1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1</pre>

- d- El formato del documento deberá respetar las siguientes características:
 - a. Hoja: A4
 - b. Márgenes: 1cm (para los 4 márgenes)
 - c. Carátula con todos los datos normales de cualquier trabajo entregable.
 - d. Tipo de letra y tamaño (para texto en general): Calibri. Tamaño: 10.
 - e. Tipo de letra y tamaño (para el código de cada programa): Courier New. Tamaño: 8
 - f. Formato o tipo de archivo: pdf
- e- Deben usar (según cada caso): bucles, condicionales, arrays o matrices, listas.
De forma obligada y en todos los programas: clases globales tanto para declaraciones como para funciones.
- f- Cualquier punto, de los anteriores mencionados, que no se cumpla en tiempo y forma tendrá como consecuencia inmediata la NO aprobación del TP en su totalidad.

Ejercicios:

- 1- Desarrollar un algoritmo que cargue una matriz de 4 x 4 con valores enteros aleatorios (Random) comprendidos entre 1 y 6. Luego de cargadas la misma, el programa debe hacer lo siguiente:
 - a- Imprimir la matriz en consola.
 - b- Recorrerla posición a posición, ir sumando el valor de cada celda y luego de terminar de recorrerla imprimir en consola, y debajo de la matriz ya impresa, el resultado de esa suma de todos los valores.
- 2- Generar un algoritmo que cargue una matriz de 5 x 5 (con el formato y valores de la imagen que se muestra a continuación mediante un algoritmo automático) y que al final de cada fila muestre la sumatoria de los valores que forman las mismas:

```
0 1 2 3 4 = 10
1 2 3 4 5 = 15
2 3 4 5 6 = 20
3 4 5 6 7 = 25
4 5 6 7 8 = 30
```

- 3- Elaborar un programa que lea o recorra una matriz, que ya debe estar cargadas con valores, y que imprima por pantalla poniendo las filas como columnas y las columnas como filas (lo que se conoce como matriz transpuesta).

Ejemplo de matriz transpuesta:

```
0 0 2 5 4
5 0 6 2 2
5 2 3 6 4

0 5 5
0 0 2
2 6 3
5 2 6
4 2 4
```

- 4- Elabore un programa que cargue una matriz de 6 x 6 con números aleatorios entre 2 y 8. Imprima la misma por pantalla. Al mismo tiempo que la imprime debe almacenar en un array de 6 posiciones los valores de la diagonal de dicha matriz (diagonal de izquierda a derecha y de arriba hacia abajo) y luego se debe imprimir dicho array.

Vea el siguiente ejemplo:

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

4 6 8 5 2 5
```

5- Desarrolle un programa que:

- Utilice dos arrays unidimensionales de 5 posiciones.
- Los arrays deberán ser cargados con valores random de entre 1 y 9.
- Luego deberá tomar cada valor del array1 e ir multiplicando por cada uno de los del array2 de la siguiente forma: $7 \times 7 \times 9 \times 3 \times 8 \times 9 = 95256$

Ejemplo:

```
Array 1: 7 8 1 5 1
Array 2: 7 9 3 8 9
Resultante: 95256 108864 13608 68040 13608
```

- 6- Desarrollar un programa que utilice un array bidimensional que permita cargar Nombre, Apellido y Edad de 5 personas (y en ese orden). Luego de que el mismo es cargado con la ayuda o interacción del usuario, el programa deberá recorrer el array e imprimir por consola el contenido de este respetando el formato que se muestra en la siguiente imagen ejemplo de ejecución:

```
Ingrese el Nombre:Leandro
Ingrese el Apellido:Pini
Ingrese la edad:43
Ingrese el Nombre:Pepe
Ingrese el Apellido:Perez
Ingrese la edad:25
Ingrese el Nombre:Lolo
Ingrese el Apellido:Polo
Ingrese la edad:18
Ingrese el Nombre:Pepa
Ingrese el Apellido:Pig
Ingrese la edad:12
Ingrese el Nombre:Nico
Ingrese el Apellido:Las
Ingrese la edad:23

Pig, Pepa 12
Polo, Lolo 18
Las, Nico 23
Perez, Pepe 25
Pini, Leandro 43
```

- 7- Desarrollar un programa que simule que se arrojan 5 dados, 5 veces, sobre la mesa y se toma nota de los valores de cada dado (la cara que queda para arriba). Los valores deben ser cargados en un Array bidimensional siendo cada fila la que representa cada una de las 5 tiradas o tiros.

Luego de estar cargado, el programa deberá recorrer el array e indicar cuantas veces salió cada cara del dado en total de todos los tiros.

Se deberá respetar el siguiente formato:

```
Presione una tecla para la jugada numero 1...
Presione una tecla para la jugada numero 2...
Presione una tecla para la jugada numero 3...
Presione una tecla para la jugada numero 4...
Presione una tecla para la jugada numero 5...

Tiro 1: 5 4 1 6 2
Tiro 2: 1 2 3 3 4
Tiro 3: 2 5 5 1 4
Tiro 4: 6 5 5 6 1
Tiro 5: 2 5 4 1 4

1 = 5 veces
2 = 4 veces
3 = 2 veces
4 = 5 veces
5 = 6 veces
6 = 3 veces
```

- 8- A los programas de los ejercicios 5 y 7 deberán incorporarle que además de imprimir por consola imprima también en un archivo de texto plano (.txt) agregando al principio la fecha y hora exacta en que se ejecuta el programa y que sea ese archivo un histórico de ejecuciones (una deba de la otra con su respectiva fecha y hora).
- 9- Desarrollar un juego (en lenguaje C# y en o de consola). El juego debe permitir que al menos dos jugadores puedan jugar de forma alternada y que al finalizar el mismo indique quien ganó.

Pre-requisitos obligatorios:

- Uso de funciones.
- Menú recursivo.
- Uso de clases globales.
- Debe llevar un registro de los resultados finales en un archivo de texto.

Ejemplo de juego:

Un programa que simule una lotería, donde genera un cartón para cada jugador con 15 números aleatorios y luego se simulen que se extraigan 15 números aleatorios de un bolillero y el jugar con más coincidencias gana.

PD:

Recuerden que estos programas serán la base de la defensa oral y práctica individual presencial que se llevará a cabo a partir del:

- 1ro A y B: jueves 31 de Octubre teniendo continuidad los días 01, 07 y 08 de Noviembre según sea necesario.
- 1ro C: lunes 04 de Noviembre teniendo continuidad el día 11 de Noviembre según sea necesario.

Con lo cual recomiendo que el desarrollo se realice de manera individual (si deciden reunirse o formar equipos pueden hacerlo, pero que no sea de más de dos personas y que cada uno use su propia lógica dado que la defensa implica realizar alguna modificación de uno de los ejercicios al azar y tendrán los minutos contados para realizarla).