

# Ejercicios Clase 4

Colecciones: Listas, Arreglos, Diccionarios, Colas, Pilas

## Ejercicios 1-10 - Obligatorios:

1. Dado que se tiene almacenado en una lista, los resultados de los últimos 10 exámenes de un alumno, calcular su promedio y mostrar por pantalla las 10 notas de los exámenes y el promedio resultante.
2. Dada las edades de 20 personas guardadas en una lista, imprimir por pantalla cuántos son mayores de edad y cuántos no.
3. Dado una lista de nombres de estudiantes, imprimir el que tenga más letras, y el que tenga menos letras de todos.
4. Crear una variable para guardar los nombres de elementos para una “lista de supermercado”. Solicitar al usuario que ingrese el nombre de un elemento que va a comprar en el super y verificar que el elemento esté en la lista. Si no está, agregarlo e indicar que no estaba. Si está, quitarlo de la lista, y avisar que sí estaba. Al finalizar mostrar por pantalla los elementos que no compró y los que compró, pero no estaban en la lista. Si se quiere, mostrar también todos los elementos que el usuario compró. Para salir el usuario debe ingresar “fin”.
5. Crear una matriz de 5 x 5. Almacenar una ‘I’ en lugares impares y una ‘P’ en lugares pares. Imprimir la matriz por pantalla
6. Se tiene una matriz de 5x7, donde 5 representa la semana de un mes y 7 los días de la semana. La estructura es para registrar la temperatura diaria de una cabina de pago, estos oscilan entre los 7 y 38 grados. Deberá llenar la matriz de forma aleatoria para el mes de mayo donde el primer día inicia en lunes y el último (31) se ubica en el miércoles (la matriz puede ser inicializada con valores aleatorios desde el principio del programa, no es necesario pedir los valores al usuario para cada posición). Se nos pide hacer lo siguiente:
  - a. Obtener la temperatura más alta y baja de la semana y que día se produjo (lunes, martes, etc.)
  - b. Promedio de temperatura de la semana.
  - c. Temperatura más alta del mes y su día.
7. Almacenar en una matriz las tablas del 1 al 9, teniendo en cuenta que en la primera fila y la primera columna se debe guardar los números (de 0 a 9), estando el cero en la primera posición (fila 0, columna 0). El resto de los lugares debe ser calculado usando los números que se dispone, por ejemplo, en la fila 1, calcular  $1*1$ ,  $1*2$ ,  $1*3$ , etc. usando las posiciones del array o arreglo. Al finalizar el cálculo, mostrar la matriz por pantalla
8. Crear una matriz de 10 x 10, y “esconder” varias ‘X’ en lugares aleatorios (la cantidad que el programador decida, pero no más de la mitad de los lugares disponibles en la matriz). El usuario deberá ingresar el lugar donde cree que hay una X, ingresando la fila y la columna por separado. Informar si acertó o no por cada ingreso. Se debe pedir al usuario ingresar valores por tantas X que se haya guardado. El usuario tiene 3 intentos para fallar. Al finalizar (Ya sea porque se

terminaron los 3 intentos, o el jugador acertó todas las X) imprimir por pantalla la matriz con sus correspondientes X, mostrando un \* donde no haya nada.

9. Diccionario de calificaciones: Crear un diccionario donde la clave sea el nombre del alumno y el valor sea su nota. El programa debe permitir:
  - a. Agregar alumnos y sus notas.
  - b. Mostrar el promedio general del curso.
  - c. Indicar el alumno con mejor nota y el de peor nota.
  - d. Hint: usar Dictionary<string, double> y recorrer con foreach
10. Simulador de atención en ventanilla: Usar una cola (Queue) para simular la atención de clientes en una ventanilla bancaria.
  - a. Encolar nombres de clientes.
  - b. Atender (desencolar) uno por uno hasta que no queden.
  - c. Mostrar en pantalla quién está siendo atendido y cuántos quedan en la fila.
  - d. Hint: usar Enqueue(), Dequeue() y Count.

## Ejercicio 11 - Extra - Opcional:

11. Inventario con múltiples colecciones: Diseñar un sistema de inventario básico usando distintas colecciones:
  - a. Un List<string> con los productos disponibles.
  - b. Un Dictionary<string, int> para registrar el stock de cada producto.
  - c. Un Stack<string> para llevar el historial de acciones realizadas (agregar, quitar, vender).
  - d. El programa debe permitir:
    - i. Agregar un producto y su cantidad.
    - ii. Vender un producto (restando stock).
    - iii. Mostrar el inventario actual.
    - iv. Mostrar las últimas 3 acciones registradas.
  - e. Hint: combina listas, diccionarios y pilas para manipular distintos tipos de información.