

Nombre:	<u>Nota</u>
Curso:	

Tiempo estimado	Tipo de actividad
-----------------	-------------------

2,5 horas

Evaluación

Normas

- El código entregado tiene que compilar sin problemas
- De cada ejercicio tendrá su propia nota, se especificará la nota mínima. Si no se alcanza esa nota mínima, la nota del examen será 4 como máximo.
- De cada ejercicio se tiene que proporcionar el main que has usado para ver el correcto funcionamiento. Si este main es escaso y se deja porciones de algoritmo sin cubrir, bajará la nota. Si es demasiado escasa, la nota máxima del examen será de 4.
- La estructura de cada fichero será el siguiente (si no, no se corregirá el ejercicio)
 - enums? delegados?
 - Una clase
 - Clases nuevas dentro de la clase
 - Atributos
 - Properties
 - Constructores
 - Destructores
 - Getters y Setters
 - Otros métodos
- Si cometes alguno de los errores comentados en clase (que son mayoritariamente 3) que no se pueden permitir, la nota máxima del examen será de 4.
- No puedes hacer trampas

(1 punto) (Nota mínima 7) Implementa la siguiente función: Implementa una función que, dado un array de enteros, te devuelva, la posición del valor mínimo, la posición del valor máximo y la mediana del array. La mediana de un array de enteros es el valor que se encuentra en el centro del array cuando sus elementos están ordenados de menor a mayor; si el array tiene un número par de elementos, la mediana es el promedio de los dos valores centrales. Puedes hacer trampas, ..., un poco

(2 puntos) (Nota mínima 7) Implementa la siguiente función.

Escribe una función que reciba una lista de números enteros y devuelva una lista de pares de números que sumen un número objetivo dado. Por ejemplo:

números = [2, 4, 3, 5, 7, 8, 9]

objetivo = 10

Resultado: [(2, 8), (3, 7)]

(2 puntos) (Nota mínima 4) Implementa la siguiente clase: Implementa una clase llamada TaskManager que gestione una lista de tareas. Cada tarea se representará con una instancia de la clase Task, y son inmutables, es decir, una vez creada una tarea no se puede cambiar.

Clase Task

- Id, nombre, descripción, fecha de realización, prioridad y estado de la tarea (que no puede ser un string)
- Con respecto a las fechas
 - DateTime dateTime1 = new DateTime(2023, 6, 15);
 - Se pueden usar los operadores normales

Clase TaskManager

- **Properties**
 - TaskCount
 - MaxDate: Devolverá la fecha mayor
- **Constructores**
 - Un único constructor que tenga el número máximo de tareas que puede almacenar.
- **Métodos**
 - Los que necesites
 - Añadir tarea
 - ContainsTask(string nombre de la tarea)
 - FilterTask(delegado): Devolverá una lista ordenada con todas las tareas que pasen el filtro. Puedes usar las funciones de sort de c#
 - ClearTasks
 - RemoveAllTasks(string): El string de esta función será el nombre de la tarea o valor de la descripción
 - Si a alguna de las funciones se le pasa algo que no es válido, deberá lanzar una excepción.

(2 puntos) (nota mínima 3) Implementa la siguiente clase, pon los atributos, constructores y métodos que veas necesarios. La clase que hay que implementar es un diccionario genérico. Entre todos los métodos y atributos que tiene que tener un diccionario para funcionar, tiene que tener 3 funciones: una que devuelva todas las keys, otra que devuelva todos los valores y otra que visite todas las entradas con un delegado.