PRÁCTICA PROGRAMACIÓN MODULAR III

Explicado en clase teoría: Terminado el tema de programación modular.

Objetivos: Seguir practicando con la implementación de funciones.

Tareas a realizar por el alumno:
(Los siguientes ejercicios están sacados del tema 6 del libro de apuntes)

1. Una aplicación almacena la siguiente información sobre cada persona: nombre, número del DNI y edad. Las personas se almacenan en un vector ordenado según el DNI. Realiza las siguientes funciones:
    - Función que, dado un número de DNI *dni* y un vector de personas, devuelva la primera posición del vector que contiene a una persona cuyo número de DNI es igual o superior a *dni* o el número de elementos del vector si no hay tal persona en el vector.
    - Función que, dado un vector de personas y una posición *p* del vector, desplace una posición a la derecha a todos los elementos del vector cuya posición sea igual o mayor que *p*.
    - Función que, dado un vector de personas y una posición *p* del vector, desplace una posición a la izquierda a todos los elementos del vector cuya posición sea mayor que *p*.
    - Función que, dado un vector de personas, muestre en la salida estándar su contenido.
    - Función que lea de la entrada estándar, y devuelva los datos de una persona. El rango de edades admitido es: *[0,150]*.

    Realiza un programa principal que permita, mediante un menú, gestionar un vector ordenado de personas. El programa debe permitir:
    - Introducir una nueva persona. Sugerencia: utilice las dos primeras funciones. Dada una nueva persona utiliza la primera función para comprobar si ya está en el vector. En caso de que ya esté se debe indicar que se rechaza su introducción. En el caso de que no esté, se tiene que crear un hueco en el vector para ella (con la segunda función) e insertar la nueva persona en la posición correspondiente.
    - Eliminar una persona. Sugerencia: utiliza la primera función para localizar su posición en el vector, si es que está, y la tercera función para suprimirla.
    - Mostrar el contenido del vector.

2. Un año es bisiesto si es divisible por 4, excepto el último de cada siglo (aquel divisible por 100), salvo que este último sea divisible por 400. De este modo 2004, 2008 o 2012 son bisiestos, pero no lo son 2100, 2200 o 2300, y sí lo es 2400. Realiza una función que dado un año devuelva si es bisiesto.

3. Define una estructura fecha que almacene el día, mes y año como enteros. Implementa funciones que permitan:
    - La lectura de una fecha correcta de la entrada estándar.
    - El envío a la salida estándar de una fecha con el formato día/mes/año.
    - La comparación de dos fechas, indicando si la primera es anterior a la segunda.
    - El incremento en un año de una fecha. Ten en cuenta los años bisiestos: el 29/2/2008 más un año es el 28/2/2009.

    Realiza también un programa principal que permita probar el funcionamiento de las funciones.

4. Una librería mantiene información de sus libros, clientes y ventas. Por cada libro se almacena un código, su título y su precio. Por cada cliente se almacena un código y su nombre. Por cada venta se almacena el código del libro y el código del cliente implicados en la venta. Se supone que en una venta un cliente compra un único libro. Se utiliza un vector para almacenar el conjunto de libros, otro vector para almacenar a los clientes y otro vector más para almacenar las ventas. Dadas estas estructuras de datos realiza funciones que permitan:
    - Dado un código de libro obtener cuántos libros con ese código se han vendido.
    - Dado un código de libro, obtener su posición en el vector de libros.
    - Obtener el título del libro más vendido.
    - Obtener el título del libro que más dinero ha recaudado.
    - Dado un código de cliente obtener un vector con los títulos de los libros que ha comprado.
    - Obtener el nombre del cliente que ha comprado más libros.
    - Obtener el nombre del cliente que ha gastado más dinero.

    Realiza un programa principal que permita probar las últimas cinco funciones.