12. Hipódromo

Desarrollar un programa que permita obtener estadísticas para el Hipódromo.

Para comenzar, se debe cargar un vector de n registros (n se ingresa por teclado). Cada registro representa una apuesta realizada para una cierta carrera, con los siguientes datos: número de ticket, caballo elegido (un número del 0 al 9, validar) y monto a apostar.

Luego, implementar un menú con las siguientes opciones:

1. Determinar el monto total de apuestas para cada caballo
2. Buscar un ticket cuyo número se ingresa por teclado. Si no existe, informarlo. Si existe, multiplicar por 10 el monto apostado y mostrar los datos de la apuesta.
3. Contar cuántas apuestas se realizaron para un cierto número de caballo, que se ingresa por teclado
4. Mostrar los datos del ticket con mayor monto apostado

14. Modelo Parcial 2019

Una compañía de servicios de limpieza desea un programa para procesar los datos de los trabajos ofrecidos. Por cada trabajo se tienen los siguientes datos: el número de identificación del trabajo, la descripción o nombre del mismo, el tipo de trabajo (un valor de 0 a 3, 0:interior, 1:exterior, 2:piletas, 3:tapizados), el importe a cobrar por ese trabajo y la cantidad de personal afectado para prestar ese servicio. Se desea almacenar la información referida a los n trabajos en un arreglo de registros de trabajos (definir el Trabajo y cargar n por teclado).

Se pide desarrollar un programa en Python controlado por un menú de opciones, que permita gestionar las siguientes tareas:  
1- Cargar el arreglo pedido con los datos de los n trabajos. Valide que el número identificador del trabajo sea positivo y que el importe a cobrar sea mayor a cero. Puede hacer la carga en forma manual, o puede generar los datos en forma automática (con valores aleatorios) o puede disponer de ambas técnicas si lo desea. Pero al menos una debe programar.  
2- Mostrar todos los datos de todos los trabajos, en un listado ordenado de mayor a menor según los importes a cobrar.  
3- Determinar y mostrar los datos del trabajo que tenga la mayor cantidad de personal afectado (no importa si hay varios trabajos con la misma cantidad máxima de personal: se pide mostrar uno y sólo uno cuya cantidad de personal sea máxima).  
4- Determinar si existe un trabajo cuya descripción sea igual a d, siendo d un valor que se carga por teclado. Si existe, mostrar sus datos. Si no existe, informar con un mensaje. Si existe más de un registro que coincida con esos parámetros de búsqueda, debe mostrar sólo el primero que encuentre.  
5- Determinar y mostrar la cantidad de trabajos por tipo.

### 7. Estudio de Arquitectura

Un estudio de arquitectos desea almacenar la información referida a sus n proyectos para sus clientes en un arreglo de registros (cargar n por teclado). Por cada proyecto se pide guardar su número de identificación, el nombre del cliente para el cual se hizo el diseño, el monto de honorarios por el proyecto, y un código numérico entre 0 y  14 para indicar el tipo de construcción a la que se sujeta el diseño (0: barrio cerrado, 1: casa de verano, 2: departamento, etc.)

Se pide desarrollar un programa en Python controlado por un menú de opciones. Ese menú debe permitir gestionar las siguientes tareas, siempre usando funciones que acepten parámetros y/o retornen valores en cada situación en que se considere apropiado:

1. Cargar el arreglo pedido. Validar que el código numérico para el tipo de proyecto esté efectivamente entre 0 y 14.

2. Mostrar todos los datos, a razón de un registro por línea en la pantalla.

3. Determinar el monto de honorarios acumulado en cada uno de los 15 tipos posibles de construcción (un acumulador para sumar los montos de los diseños tipo 0, otro para los proyectos tipo 1, etc.)

4. Muestre todos los proyectos cuyo código de tipo de proyecto sea diferente de 4. Este listado debe salir ordenado de menor a mayor, de acuerdo al monto de honorarios.

5. Determinar si existe algún diseño para el cliente cuyo nombre sea igual a x, siendo x una cadena que se carga por teclado. Si existe, mostrar todos los datos de ese diseño por pantalla. Si no existe, informar con un mensaje.

8. Concursos Administracion Pública

El gobierno de la provincia de Córdoba desea guardar la información referida a los resultados de los exámenes de concursos por cargos en la administración pública, en un arreglo de *n*registros (donde *n*es un valor que se carga por teclado). Por cada resultado del concurso, se pide guardar el dni del concursante, su nombre, el cargo para el que se postuló (un código que va de 0 a 19, o sea, hay 20 cargos) y el puntaje obtenido (un valor de 0 a 100 , que puede tener decimales).

Se pide desarrollar un programa en Python controlado por un menú de opciones. Ese menú debe permitir gestionar las siguientes tareas a partir del arreglo pedido en el párrafo anterior:

1. Cargar el arreglo pedido con los datos de los *n*resultados. Validar que el código del cargo se encuentre entre 0 y 19.
2. Mostrar los datos de los concursantes que hayan aprobado el examen (se aprueba con 70 puntos o más).
3. Determinar cuántos concursantes rindieron el examen por cada tipo de cargo (es decir, cuántos concursantes rindieron por el cargo 0, cuántos por el cargo 1, cuántos por el cargo 2, etc... hasta el cargo 19).
4. Mostrar los datos del arreglo, ordenados de mayor a menor, por el puntaje obtenido en el examen.
5. Cargar por teclado el nombre de un postulante, y mostrar por pantalla todos los datos del mismo si se encuentra en el vector. Si este postulante además aprobó el concurso, muestre un mensaje que resalte ese resultado además de sus datos. Informe con otro mensaje si el postulante no se encuentra en el vector.