



## Práctica Final. Opción 1: Sistema de gestión y cálculo de estadísticas de combinaciones de la lotería primitiva

Fecha de publicación: Martes 14/12/2021

Fecha de entrega: Lunes 24/01/2022

*Este programa vale el 70% de la nota del laboratorio*

*Se valorará la elección de las estructuras de datos más convenientes, así como la adecuada división del programa en funciones. Recordar también que se valorará la utilización correcta de la Guía de estilo de C/C++ y el diseño de una correcta interfaz de usuario.*

Se pretende realizar un programa para la gestión y el cálculo de estadísticas de combinaciones de la lotería primitiva. Para ello se van a establecer una serie de requisitos mínimos que el programa debe tratar de satisfacer. Dichos requisitos son los siguientes:

1. Los datos de interés de las combinaciones ganadoras son: fecha del sorteo, los 6 números de la combinación ganadora, el número complementario de la combinación ganadora y el reintegro. Los números del sorteo, así como el número complementario, están en el rango 1..49, mientras que los números de reintegro están en el rango 0..9.
2. Las combinaciones estarán almacenadas en un fichero de texto que el programa deberá **leer y cargar en memoria** al inicio del programa para el trabajo con estadísticas. Asimismo, el programa deberá poder modificar dicho fichero, a petición del usuario o al finalizar la ejecución del programa, para reflejar actualizaciones sobre las combinaciones que se puedan realizar.
3. El programa deberá contener una separación lógica de funcionalidades, organizadas a través de menús, de forma que el menú principal tenga 3 opciones:
  - a. Actualización de combinaciones
  - b. Cálculo de estadísticas
  - c. Salir
4. El menú de actualización de combinaciones debe poder proporcionar las siguientes posibilidades:
  - a. Buscar una combinación por fecha.
  - b. Visualizar todas las combinaciones entre dos fechas concretas.
  - c. Comprobación de combinaciones. Esta función permitirá introducir 6 números en el programa, y el programa deberá comprobar el número de aciertos obtenidos en todas las combinaciones almacenadas, mostrando aquellas combinaciones en las que se han conseguido más de 3 aciertos, indicando además el número de aciertos obtenidos.
  - d. Insertar una combinación. La inserción debe solicitar todos los datos necesarios para una combinación. Los 6 números de la combinación no tienen porque introducirse en orden ascendente, pero deberán quedar almacenados en dicho orden. La inserción debe mantener el orden de las combinaciones por fecha. No se debe poder insertar una combinación que ya exista en el sistema. En este punto, las inserciones **sólo se realizará en memoria, y no en el fichero.**



- e. Eliminar una combinación por fecha. La función de eliminación solicitará la fecha del sorteo para eliminar la combinación. El programa debe alertar de intentos de borrado de combinaciones inexistentes. En este punto, las eliminaciones **sólo se realizarán en memoria, y no en el fichero**.
  - f. Actualizar el fichero de datos. Esta opción volcará el estado de las combinaciones de memoria en el fichero de datos correspondiente, actualizando completamente todos los datos a través de las acciones previas que se hayan realizado.
5. El menú de cálculo de estadísticas debe proporcionar las siguientes opciones:
- a. Obtener los  $x$  ( $x$  entre 1 y 49) números que más frecuentemente aparecen en las últimas  $y$  combinaciones, ordenados por frecuencia.
  - b. Obtener los  $x$  ( $x$  entre 1 y 49) números que menos frecuentemente aparecen en las últimas  $y$  combinaciones, **considerando los números que no aparecen** en las combinaciones, ordenados por frecuencia.
  - c. Obtener una frecuencia de aparición en las combinaciones de 2 números consecutivos para todas las parejas de números consecutivos posibles (1-2, 2-3, ..., 48-49) en las últimas  $x$  combinaciones, ordenados por frecuencia.
  - d. Obtener las frecuencias de los reintegros para los 10 posibles valores en las últimas  $x$  combinaciones.
  - e. Generar una combinación completa de forma aleatoria.
  - f. Generar una combinación de forma automática teniendo en cuenta la información estadística que se puede obtener del programa, y conteniendo al menos dos números consecutivos.
6. Al salir del programa debe realizar lo siguiente:
- a. Preguntar si se desean guardar los cambios en el fichero de datos (solo si estos han sido modificados y no han sido guardados previamente) y obrar en consecuencia.
  - b. Cerrar el programa

Para facilitar las tareas de inserción de datos en el programa, se proporciona un fichero (**Primitiva.dat**) que contiene los datos que el programa necesita con las combinaciones ganadoras de sorteos realizados durante el año 2021. Este fichero debe ser la base para el inicio del trabajo, y debe poder ser modificado desde el programa. Se recomienda que se mantenga la estructura del fichero para su procesamiento (campos separados por tabuladores con 2 primeras líneas explicativas de los contenidos).

Las funciones que se solicitan en esta práctica son requisitos mínimos. El cumplimiento de estos requisitos se considera obligatorio para una valoración positiva del ejercicio. También se pueden realizar las ampliaciones que se consideren oportunas, siempre que tengan lógica en el contexto de la aplicación. Estas mejoras no serán valoradas en absoluto si los requisitos mínimos no se cumplen o no funcionan correctamente.



### Realización estructurada de la práctica:

1. Diseña una estructura de datos adecuada para guardar los 6 números premiados de la lotería primitiva.

Diseña también una estructura de datos adecuada para guardar una fecha (día, mes y año).

Utilizando las estructuras de datos anteriores, diseña una estructura de datos en la que se pueda guardar la información completa de **UN** sorteo (fecha, números premiados, complementario y reintegro).

Diseña una estructura de datos donde poder guardar tanto la información de todos los sorteos como el número de sorteos guardados en la estructura de datos.

2. Realiza un programa completo que, utilizando una función, lea toda la información contenida en el fichero, si ha podido abrir el fichero, y, utilizando otra función, la muestre por pantalla. Es **MUY** recomendable que te apoyes en funciones auxiliares tales como una función que lea **UN** solo sorteo de un fichero ya abierto o una función capaz de mostrar por pantalla **UN** solo sorteo.
3. Modifica el funcionamiento del programa anterior que muestra las combinaciones por pantalla para que guarde toda la información en un fichero. Utiliza una función para guardar la información en el fichero. A la función le pasaremos el nombre del fichero y la información y **NO** mostrará nada por pantalla, sólo guardará la información en el fichero, devolviendo al programa principal si ha podido o no abrir el fichero. Es **MUY** recomendable que te apoyes en una función auxiliar que guarde **UN** solo sorteo en un fichero ya abierto.

Realiza las siguientes funciones:

4. Realiza una función que compare dos fechas y nos devuelva 0 si son iguales, -1 si la primera es anterior a la segunda y 1 si la primera es posterior a la segunda. Utiliza para hacer la comparación una adaptación a 'fechas' de la función **ConvierteHoraNum** vista en la práctica 5.
5. Utilizando esa función y modificando adecuadamente la función 'Buscar en vector ordenado' que está en los apuntes del profesor, realiza una función que nos diga en que posición se encuentra un determinado sorteo a partir de una determinada fecha que se pasa como parámetro.
6. Realiza una función a la que se le pasen dos combinaciones de números y nos devuelva cuantos de los números de la primera combinación están contenidos en la segunda.
7. Realiza una función que nos diga cuantas veces aparece cada uno de los números de las combinaciones ganadoras de los sorteos leídos del fichero (cuantos veces aparece el uno, cuantas veces el dos, ... y cuantas el 49). Utiliza un vector con 49 o 50 posiciones para contar el número de apariciones de cada valor.
8. Modifica adecuadamente la función anterior para que acepte un entero 'num' y nos diga el número de veces que aparece cada uno de los valores de las combinaciones en las 'num' combinaciones más recientes.

Ninguna de las funciones realizadas entre los puntos 3 y 8 mostrarán **NADA** por pantalla **NI** pedirán **NADA** por teclado.

9. Realiza una función a la que se le pase un vector de 50 enteros y nos muestre por pantalla el mayor elemento contenido en el vector.
10. Modifica adecuadamente el ejercicio anterior para que se muestre primero el mayor elemento y a continuación el segundo mayor.



11. Modifica adecuadamente el ejercicio anterior para que se pase a la función además del vector con 50 enteros, otro entero 'num' y nos muestre por pantalla en orden decreciente los 'num' enteros mayores.

Una vez realizadas y probadas estas funciones, procede a ajustarlas e integrarlas, así como a desarrollar las que creas conveniente para poder realizar el programa completo.