

Trabajo I - Juego del Sudoku

Nota Máxima: 7 puntos

El trabajo consiste en hacer un programa para jugar al sudoku.

Los sudokus deben estar guardados en un archivo de texto con el siguiente formato

```

--849-6--
6-47-32--
-----7
-8-2--1--
--1--857-
7-65-43-9
86-----
-1--2--58
-4--5----
```

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 8 | 4 | 9 | | 6 | | |
| 6 | | 4 | 7 | | 3 | 2 | | |
| | | | | | | | | 7 |
| | 8 | | 2 | | | 1 | | |
| | | 1 | | | 8 | 5 | 7 | |
| 7 | | 6 | 5 | | 4 | 3 | | 9 |
| 8 | 6 | | | | | | | |
| | 1 | | | 2 | | | 5 | 8 |
| | 4 | | | 5 | | | | |

El programa debe:

- Tener 5 sudokus guardados en 5 archivos.
- Cargar y visualizar cada vez que se juega un sudoku aleatoriamente.
- Permitir al usuario elegir qué fila y columna quiere rellenar
- Completar el sudoku con la fila/columna y el valor elegido.
- Cada vez que se completa un cuadrante/fila/columna se debe comprobar que están todos los números del 1 al 9
- Si el usuario completa una fila/columna/cuadrante de modo incorrecto se acaba la partida mostrando el mensaje "Error" (utilizando excepciones).
- Cuando estén completas todas las casillas sin haber infringido las reglas del juego finaliza el programa mostrando el mensaje "Ganador".

En el trabajo debéis al menos:

- Utilizar clases.
- Utilizar sobrecarga.

Entregables:

- Código fuente desarrollado (80% de la nota)
- Memoria descriptiva del trabajo realizado (20% de la nota)

Trabajo II - Resuelve Sudoku

Nota Máxima: 10 puntos

El trabajo consiste en hacer un programa que resuelva automáticamente un SUDOKU.

El SUDOKU debe estar guardado en un archivo de texto con el siguiente formato

--849-6--
6-47-32--
-----7
-8-2--1--
--1--857-
7-65-43-9
86-----
-1--2--58
-4--5----

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 8 | 4 | 9 | | 6 | | |
| 6 | | 4 | 7 | | 3 | 2 | | |
| | | | | | | | | 7 |
| | 8 | | 2 | | | 1 | | |
| | | 1 | | | 8 | 5 | 7 | |
| 7 | | 6 | 5 | | 4 | 3 | | 9 |
| 8 | 6 | | | | | | | |
| | 1 | | | 2 | | | 5 | 8 |
| | 4 | | | 5 | | | | |

El programa debe

- Pedir la ruta del archivo donde está guardado el sudoku.
- Cargarlo.
- Visualizarlo por pantalla sin resolver.
- Resolverlo y mostrar por pantalla la solución.

En el trabajo debéis al menos:

- Utilizar clases.
- Utilizar herencia (tablero | sudoku)
- Utilizar sobrecarga.

Entregables:

- Código fuente desarrollado (80% de la nota)
- Memoria descriptiva del trabajo realizado (20% de la nota)

Trabajo III - Demo heliocentrista

Nota Máxima: 10 puntos

Implementar una representación gráfica del movimiento relativo entre el sol, la tierra y la luna.

En el trabajo debéis al menos:

- Buscar librería C++ adecuada para el ejercicio (puedes usar la que consideres pero Glut es una buena opción)
- Instalar la librería en tu computador.
- Utilizar clases.
- Utilizar herencia (ej: cuerpo celeste | astro | planeta | satélite).

Entregables:

- Código fuente desarrollado (80% de la nota)
- Memoria descriptiva del trabajo realizado (20% de la nota)
- Vídeo de captura de pantalla en el que se vea el funcionamiento del programa.

Trabajo IV - Grafo

Nota Máxima: 10 puntos

Implementar un grafo que represente una red de amigos. Dicho grafo debe incorporar los métodos que consideres necesarios para su uso como red de amistad, un método que devuelva true si las personas pasadas como argumento son amigos directos, un método que devuelva un contenedor con todas las personas del grafo y un método que devuelva un entero indicando su grado de separación (si dichas personas no tienen relación directa el método debe lanzar una excepción)

En el trabajo debéis al menos:

- Utilizar clases.
- Utilizar sobrecarga.

Entregables:

- Código fuente desarrollado (80% de la nota)
- Memoria descriptiva del trabajo realizado (20% de la nota)
- Vídeo de captura de pantalla en el que se vea el movimiento de los planetas.

Trabajo V - ¡A luchar!

Nota Máxima: 10 puntos

Implementar un programa que permita pelear a 2 equipos de personajes. Los personajes disponibles serán bárbaros y magos.

En el trabajo debéis:

- Utilizar clases.
- Utilizar herencia.
- Utilizar polimorfismo.
- Utilizar sobrecarga.
- Crear 2 equipos de 10 personajes aleatorios, teniendo cada uno de los personajes características aleatorias (vida, ataque simple, defensa, ataque mágico, ataque bárbaro).
- El programa de manera cíclica debe permitir mediante un menú textual:
 - La creación de los 2 equipos
 - Comenzar la batalla

Entregables:

- Código fuente desarrollado (80% de la nota)
- Memoria descriptiva del trabajo realizado (20% de la nota)