

### **Driver Ctrl Rankings:**

#### Prueba 1

- Descripción: test de las funciones Set() y test de las funciones Get() de la clase *CtrlRanking*.
- Objetivos: comprobar que funcionan bien los gets y los sets de esta clase
- Drivers: driverCtrlRanking
- Entrada: 1; 2; 3 1 marti 38,87 4 3; 4 5; 5 5
- Salida: resultado de los gets, el parámetro correspondiente queda actualizado
- Resultado de la prueba: ok

Casos de prueba:(siempre primero se debe hacer la opción 1 que es la que carga los rankings, sino las otras funciones no funcionan ya que no hay rankings sobre los que operar)

- entrada = 2
- salida = 6
  
- entrada = 3 1 marti 38,87 4 3
- salida = x 5.0  
a 5.0  
a 9.6  
marti 10.0  
ruben 20.5  
marti 38.87
  
- entrada = 4 5
- salida = tito 198.43  
manuel 201.34  
josepet 215.6  
mariet 289.63
  
- entrada = 5 5
- salida = tito 198.43  
manuel 201.34  
josepet 215.6

### **Driver Ctrl Usuarios:**

#### Prueba 1

- Descripción: test de las funciones Set() y test de las funciones Get() de la clase *CtrlRanking*.
- Objetivos: comprobar que funcionan bien los gets y los sets de esta clase
- Drivers: driverCtrlRanking
- Entrada: 1; 2; 3; 4 marti; 5 a 3; 6 a 1 3; 6 a 1; 7 a 1; 7 a 2; 8 marti 123 4 marti

- Salida: resultado de los gets, el parámetro correspondiente queda actualizado
- Resultado de la prueba: ok

Casos de prueba:(sempre primer s'ha de fer l'opció 1 que és la que carga els usuaris, sinó les altres funcions no funcionen ja que no hi ha usuaris sobre els que operar)

- entrada = 2
- salida = 2
  
- entrada = 3
- salida = marti  
a
  
- entrada = 4 marti
- salida = marti 1234
  
- entrada = 5 a 3
- salida = marti
  
- entrada = 6 a 1 3
- salida = marti  
a
  
- entrada = 6 a 1
- salida = Ya existe ese Usuario
  
- entrada = 7 a 1
- salida = contraseña correcta
  
- entrada = 7 a 2
- salida = contraseña incorrecta
  
- entrada = 8 marti 123 4 marti
- salida = marti 123

### **Driver Pair:**

#### Prueba 1

- Descripción: test de las funciones Set() y test de las funciones Get() de la clase *Pair*.
- Objetivos: comprobar que funcionan bien los gets y los sets de esta clase
- Drivers: driverPair
- Entrada: 3 ruben 1; 4 29,45 2
- Salida: resultado de los gets, el parámetro correspondiente queda actualizado
- Resultado de la prueba: ok

Casos de prueba:

- entrada = 3 ruben 1
- salida = ruben
- entrada = 4 29,45 2
- salida = 29.45

### **Driver Ranking:**

Prueba 1

- Descripción: test de las funciones Set() y test de las funciones Get() de la clase *Ranking*.
- Objetivos: comprobar que funcionan bien los gets y los sets de esta clase
- Drivers: driverRanking
- Entrada: 3 ruben 58,456 3 marti 38,972 1; 4 ruben 58,456 4 marti 38,972 1; 3 ruben 58,456 3 marti 38,972 3 felix 29,56 3 victor 39,27 5 2
- 
- Salida: resultado de los gets, el parámetro correspondiente queda actualizado
- Resultado de la prueba: ok

Casos de prueba:

- entrada = 3 ruben 58,456 3 marti 38,972 1
- salida = marti 38,972  
ruben 58,456
- entrada = 4 ruben 58,456 4 marti 38,972 1
- salida = marti 38,972  
ruben 58,456  
(es guarda al fitxer .txt corresponent)
- entrada = 3 ruben 58,456 3 marti 38,972 3 felix 29,56 3 victor 39,27 5 2
- salida = felix 29,56  
marti 38,972  
victor 39,27

### **Driver Usuario:**

Prueba 1

- Descripción: test de las funciones Set() y test de las funciones Get() de la clase *Usuari*.
- Objetivos: comprobar que funcionan bien los gets y los sets de esta clase
- Drivers: driverUsuario

- Entrada: 3 marti 1; 4 1234 2; 5 marti 1234; 5 marti 123
- Salida: resultado de los gets, el parámetro correspondiente queda actualizado
- Resultado de la prueba: ok

Casos de prueba:

- entrada = 3 marti 1
- salida = marti
- entrada = 4 1234
- salida = 1234
- entrada = 5 1234
- salida = contraseña correcta
- entrada = 5 123
- salida = contraseña incorrecta

### **Driver SolverValida:**

Prueba 1

- Descripción: test de la funciones de validador y solucionador
- Objetivos: comprobar que funcionan bien el validador y solucionador
- Drivers: driverSolver
- Entrada: números correspondientes a las operaciones
- Salida: el tablero resuelto y el número de soluciones
- Resultado de la prueba: ok

Casos de prueba:

- entrada = 1 1 1 -1 2 1 -1 3 1 -1 4 1 -1 5 1
- salida =  
Este Kakuro tiene una solución única.  
\* \* C19 C12 \* \* \* C7 C10  
\* F14 5 9 C4 C11 C17F4 1 3  
\* C7F36 2 3 1 8 9 6 7  
F12 4 8 F10 3 2 5 C25 C14  
F3 2 1 C20 C11F20 1 3 9 7  
F17 1 3 9 4 C8 F6 1 5  
\* C11 C7F13 7 1 5 C4F10 8 2  
F28 7 5 4 6 3 1 2 \*  
F6 4 2 \* \* F8 3 5 \*

Este Kakuro tiene una solución única.

\* C23 C38 C3 C10 \* \* C11 C13 \*

F26 8 9 2 7 C3 F4 3 1 C18  
 F15 4 5 1 3 2 C27F11 8 2 1  
 F4 1 3 C16 C38F10 1 9 C14F12 3 9  
 F29 5 8 9 7 C21F29 5 9 7 8  
 F28 2 6 7 3 4 1 5 C22 C38  
 F10 3 7 C10F23 8 9 6 C9F4 1 3  
 \* C20 C10F34 9 6 8 2 1 3 5  
 F11 3 2 1 5 C5F28 4 8 7 9  
 F12 9 3 C12F11 9 2 C4 C3F10 2 8  
 F21 8 4 9 F16 3 1 2 4 6  
 \* F4 1 3 \* F16 3 1 5 7

Este Kakuro tiene una solución única.

\* \* \* \* C35 C17 \* \* \* C17 C4 \* \* \* \*  
 \* \* \* F17 9 8 C3 \* C10F12 9 3 \* \* \* \*  
 \* \* \* C16F17 7 9 1 C17F12 3 8 1 \* \* \* \*  
 \* \* F15 9 6 F14 2 8 4 \* \* C16 C7 \* \*  
 \* \* F12 7 5 C35 C3F11 9 2 C17 F8 7 1 \* \*  
 \* C23 C11 C4F18 8 9 1 C23F9 1 8 C16F11 9 2 C30 \*  
 F11 6 2 3 C16F13 5 2 6 C6F10 9 1 C16F10 4 6 C7  
 F28 9 5 1 7 6 F10 8 2 F11 2 9 C4F12 8 4  
 F11 8 3 C24F16 9 7 C16F12 9 3 C16F23 4 7 1 9 2  
 \* F8 1 7 C4F15 8 7 C11F16 1 9 6 C35F11 3 7 1  
 \* \* F9 8 1 F11 9 2 C3F18 7 3 8 C4 \* \*  
 \* \* F12 9 3 \* F3 1 2 C3 F6 5 1 \* \*  
 \* \* \* \* C17 C3F8 5 1 2 C17F10 7 3 \* \*  
 \* \* \* \* F14 9 2 3 F18 1 8 9 \* \* \*  
 \* \* \* \* F9 8 1 \* \* F15 9 6 \* \* \*

Este Kakuro tiene más de una solución

\* \* C12 C16 \* \*  
 \* C10F11 5 6 \* \*  
 F8 1 3 4 C15 C14  
 F25 9 4 1 3 8  
 \* \* F13 2 5 6  
 \* \* F10 3 7 \*

Este Kakuro no tiene solución!

\* C10 C15 \* C10 C15  
 F4 ? ? F8 ? ?  
 F3 ? ? F6 ? ?  
 F8 ? ? F9 ? ?

F6 ? ? F3 ? ?  
\* \* \* \* \*