

UNIVERSIDAD DE SANTIAGO DE CHILE

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INFORMÁTICA

**MANUAL DE USUARIO**

**SUDOKU**

**Programado en C.**

**POR: RODRIGO CERDA R.**

**PROFESOR: ROBERTO GONZALEZ I.**

**Santiago, 24 de Noviembre 2015**

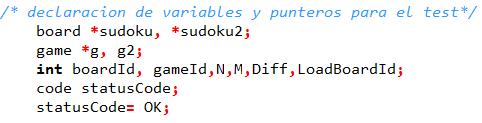
En el siguiente manual, se pretende explicar la manera de operar de las funciones generadas en el programa, además de ejemplos de cómo implementarlas.

Consideraciones:

* El proyecto fue probado en dev-c++
* Para esta representación, las coordenadas estan entre [0,tamañoMáximo[.
* Para las funciones de guardado y lectura de archivos, se asume la existencia de las carpetas contenedoras.

Las funciones a probar están implementadas en el archivo test.c, comentadas claramente para su uso.

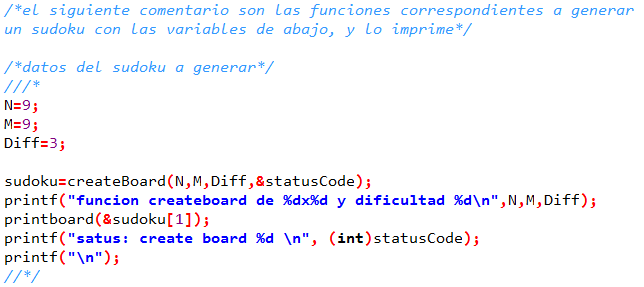
Variables declaradas en test.c

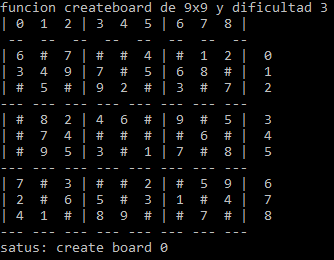


El puntero \*sudoku es el encargado de recibir el resultado de createboard.

Como el resultado de dicha función es un puntero a dos boards con memoria ya apartada, no es necesario darle memoria en el main.

CreateBoard:





Recalcar que createboard entrega dos estructuras board, el primero, [0], es la estructura inicial, y [1] es en el que se jugará.

Printboard:

Función que solicita un puntero a board y lo imprime en pantalla, como en el ejemplo anterior.

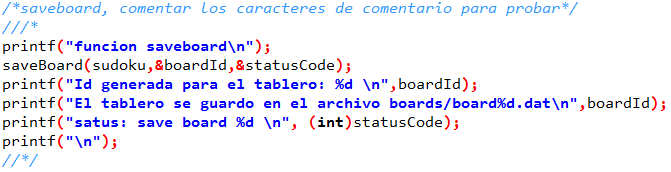
Si bien, createboard define un arreglo de dos boards, printboard necesita como variable de entrada un puntero a solo un board, por lo que para imprimir el resultado de createboard hay que especificar que board se quiere imprimir.

Puntero al segundo sudoku

Para efectos de test.c, en los ejemplos se solicita imprimir siempre el sudoku [1] , que es el el que el usuario debería ver.

Saveboard:

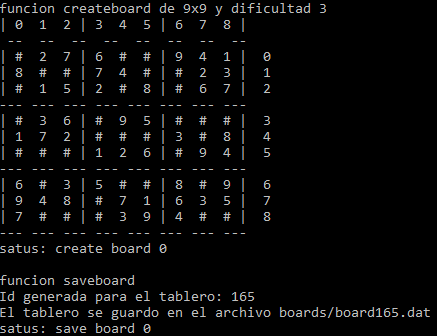
Función sin retorno, que se encarga de almacenar en un fichero el tablero generado en la función createboard.



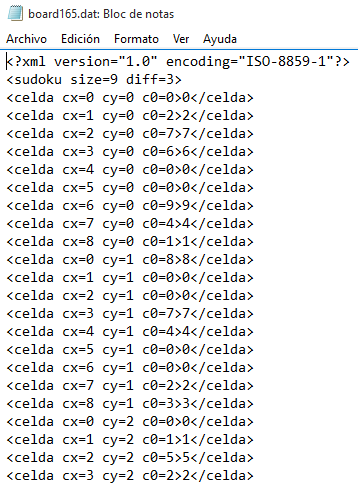
Esta función almacena ambos sudoku, (el que incluye el tablero inicial, y el que incluye el tablero jugable). Por lo que el puntero a board que se solicita como parámetro de entrada debe apuntar a dos board.

Luego de aplicar la función se imprimen en pantalla datos de su ejecución, para comprobar lo guardado dentro del fichero especificado.

Ej:

Si uno revisa el fichero board165.dat,

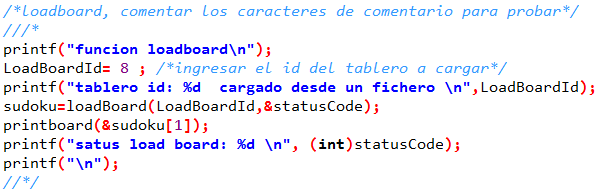
Gracias a la implementación XML, resultará fácil confirmar que el tablero generado es el que ha sido almacenado.

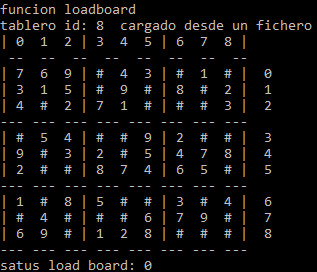


Notar que el atributo c0 del nodo celda es igual al elemento contenido dentro de ellas, esto es porque en un principio ambas dimensiones son iguales.

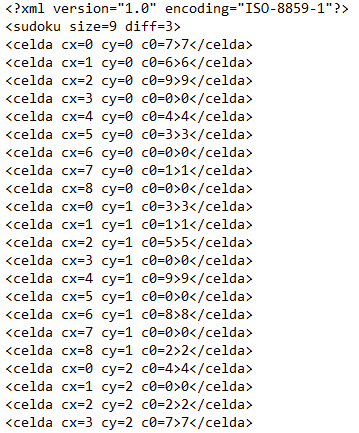
LoadBoard:

Esta función entrega un puntero a dos boards, pero por el hecho de generar el espacio en memoria dentro de su llamado, no es necesario generar ese espacio en el main. Basta con asignarle el resultado al puntero \*sudoku.





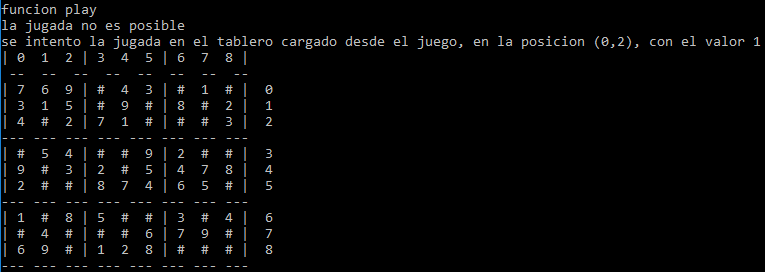
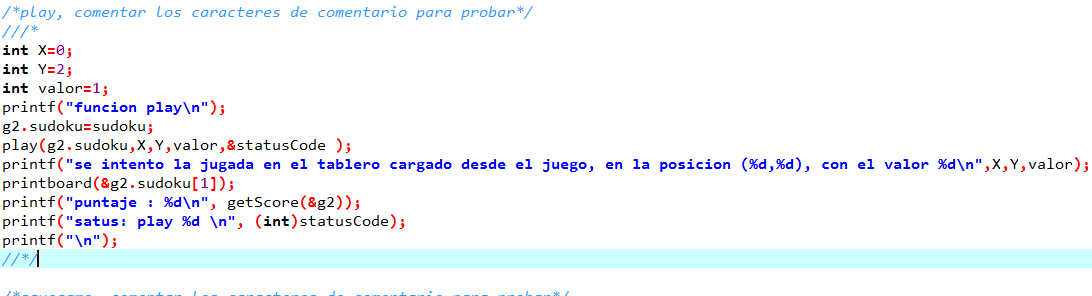
Para confirmar su efectividad, al igual que en saveboard, se podría abrir el fichero board8.dat para comparar el resultado.



Play:

Recibe un puntero a dos board, las coordenadas y el valor a insertar.

Dentro de la función, analiza que el board[0] contenga un cero en la posición a jugar, para así permitir dicha jugada en el board[1].

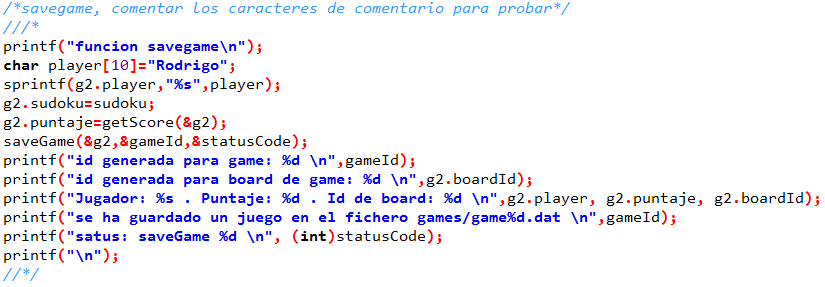


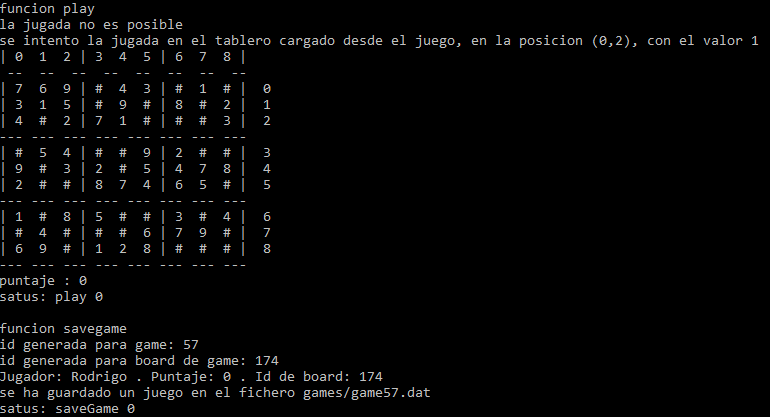
En este caso, la jugada no fue posible por tener un numero distinto de cero dentro del board[0], en las coordenadas (0,2).

SaveGame:

Para guardar un game, primero hay que añadirle datos. Para eso se utilizara la variable game g2.

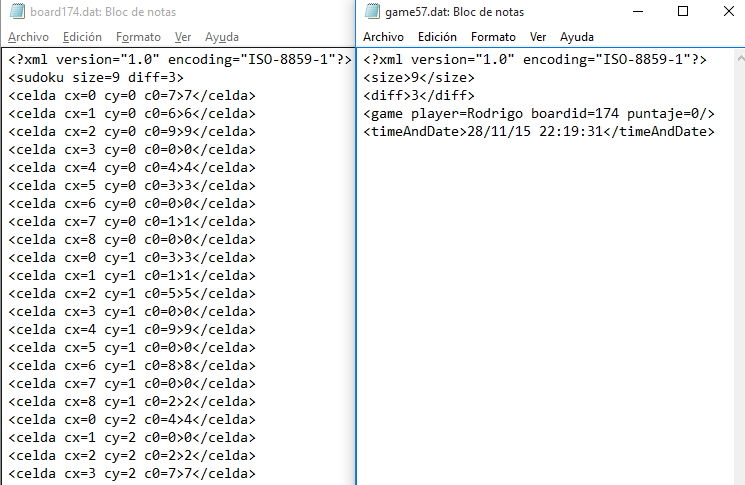
Notar que dato sudoku que almacena la estructura, es un array de 2 board, por lo que se le debe asignar sudoku





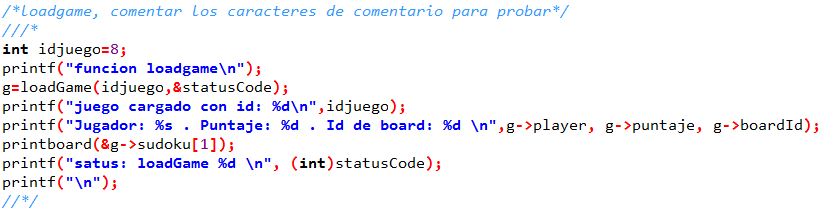
El fichero board creado al llamar a savegame es el numero 174, por lo que al abrir dicho archivo se debería poder comparar satisfactoriamente lo guardado con el tablero de consola.

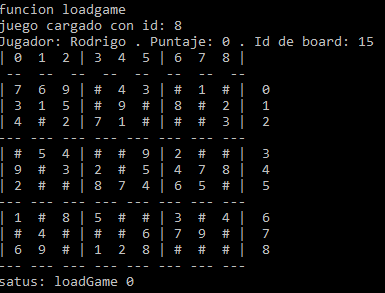
El fichero game generado es el numero 57, por lo que en el deberían verse el resto de los datos solicitados.



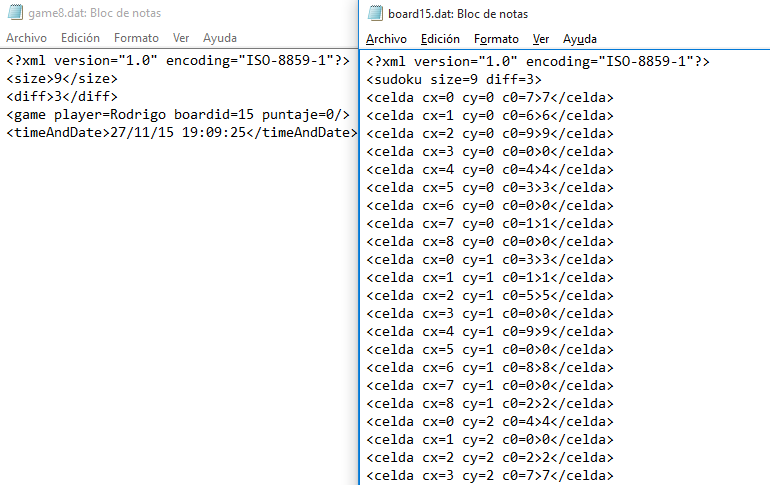
LoadGame:

Entrega un puntero a una estructura game cargado desde un fichero con un id particular, con memoria generada en el momento del llamado a esta función, por lo que para recibir el resultado se emplea “g”, que está declarado como un puntero a game.



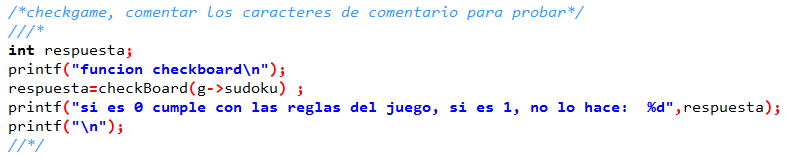


Para comprobar su efectividad, nuevamente se podrían comparar los datos impresos en consola, con los incluidos en los ficheros board15.dat y game8.dat



Checkgame:

Recibe un puntero a dos boards, evaluando el board[1] para entregar 0 en caso de que sus celdas cumplan las reglas del juego, y 1 en caso contrario.



En las funciónes saveboard y play, se esta empleando la función designada como extra “getscore”, que entrega un int con el porcentaje de tablero completado, almacenándolo en caso de saveboard.