
From Programacion 1

Laboratorio: Tarea2

Segunda Tarea. Programación 1

Para obtener una versión apropiada para impresión, siga el vínculo *print* en la esquina superior derecha.

Tabla de Contenidos

Ultima modificación: 21 octubre de 2005, hora 14:27

- **Introducción**
- **Información General**
- **La Tarea. Resolver un sudoku**
- **Terminología**
 - **Fila, Columna y Región**
 - **Candidato de una celda**
- **Métodos de resolución**
 - **Candidato Único**
 - **Candidato Exclusivo**
 - **Procedimiento General**
- **Se pide**
 - **La entrada del programa**
 - **La salida del programa**
- **Se puede utilizar**
- **Ejemplos de sesiones**

Introducción

Este documento presenta el problema que deberá resolverse para la aprobación de la segunda tarea del laboratorio del curso 2005: *un programa que resuelve un sudoku*. Contiene información general acerca del laboratorio y la letra del problema a resolver.

Contiene información acerca de: normas, recursos, plazos, una presentación general del problema, las especificaciones del mismo, ejemplos de ejecución y sobre la forma de entrega.

Información general

El estudiante que no respete alguna de las consideraciones que siguen, corre el riesgo de que su trabajo sea invalidado, con la consiguiente reprobación del curso.

Compilador

Todos los programas deben ser compatibles con el compilador del curso (**Turbo Pascal 5.5**). No se aceptarán como válidas aquellas tareas que pudieran funcionar con algún otro compilador Pascal, pero no funcionen con TP55

Individualidad

La tarea es **individual**. Los estudiantes pueden estudiar en grupo y consultarse mutuamente, pero **no deben realizar en grupo** las tareas de *codificación, tipeo, compilación y depuración* del programa. Los trabajos que **a juicio de los docentes** no cumplan con estos requisitos serán eliminados y se comunicará al Consejo para que tome las sanciones que correspondan.

Calendario

La tarea puede entregarse en la semana que va del 21/11 al 25/11. Se aceptan entregas hasta el viernes 25/11 a la hora 24.

Los trabajos deberán ser entregados dentro de los plazos establecidos. **No se aceptarán trabajos fuera de plazo bajo ningún concepto.**

Forma de entrega

Las entregas se realizarán por la *web*. Para ello se deberá acceder a un sitio destinado a tal fin y seguir los pasos que se explicarán en la página correspondiente. Esta página estará disponible cuando comience el plazo de entrega.

La tarea. Resolver un *sudoku*

Sudoku es un puzzle lógico cuyo objetivo es rellenar una cuadrícula de 9×9 celdas dividida en regiones de 3×3, con las cifras del 1 al 9, a partir de algunos números ya dispuestos en algunas de las celdas. No se debe repetir ninguna cifra en una misma fila, columna o región. Un sudoku está bien planteado si la solución es única (www.wikipedia.org/).

En la figura aparece un sudoku

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

Y aquí tenemos la solución:

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

Terminología.

Fila, Columna y Región

Consideraremos un sudoku como una matriz de 9x9 y utilizaremos los términos **fila** y **columna** con su significado habitual. Una fila es una secuencia de 9 celdas contiguas alineadas horizontalmente. Una columna es una secuencia de 9 celdas contiguas alineadas verticalmente. Utilizaremos los números del 1 al 9 para identificar las filas y columnas. Las filas se numeran de izquierda a derecha, las columnas de arriba hacia abajo. Una celda se identifica por un par (i, j) donde i representa la fila y j la columna.

Utilizaremos el término **región** para referirnos a cada una de las 9 subcuadrículas de 3x3. Las regiones se numeran de 1 a 9 de izquierda a derecha y de arriba hacia abajo. Así por ejemplo la región 2 contiene el conjunto de celdas:

$$\{(i, j) \mid 1 \leq i \leq 3, 4 \leq j \leq 6\}$$

Candidato de una celda

Dado un sudoku, un número $x \in \{1,2,3,4,5,6,7,8,9\}$ es un **candidato** para la celda (i,j) vacía, si se cumple que x no aparece en la fila i , ni en la columna j , ni en la región que contiene a la celda (i,j) .

Notar que candidato sólo se define para las celdas vacías.

$C(S,i,j)$ representa el conjunto de candidatos de la celda (i,j) en el sudoku S .

Así por ejemplo, si llamamos $S1$ al sudoku de la primera figura, tendremos:

- $C(S1,1,1) = \{2,8,9\}$
- $C(S1,9,9) = \{9\}$

Métodos de resolución

El sudoku puede resolverse fácilmente por métodos computacionales conocidos como **ensayo y error**. A grandes rasgos en estos métodos se elige para una celda aún vacía, uno cualquiera de sus candidatos y se continúa con las restantes celdas. Si en algún momento se llega a una configuración que no tiene solución, se deshace la última elección y se elige otro candidato para esa celda. Se continúa así hasta agotar todas las posibilidades o encontrar alguna solución.

En esta tarea no aplicaremos métodos de ensayo y error sino que nos limitaremos a los métodos lógicos simples que se describen a continuación.

Candidato único

Decimos que un dígito d es candidato único para una celda (i,j) en un sudoku S si se cumple que:

$$C(S,i,j) = \{d\}$$

Cuando esto se cumple podemos asegurar que la celda (i,j) debe ser ocupada por el dígito d .

En el sudoku de la figura, la posición **(9,9)** tiene como candidato único el **9**

Candidato exclusivo

Decimos que un dígito d es un candidato exclusivo de una celda (i,j) , si d es candidato para esa celda y además se cumple al menos una de las siguientes condiciones:

- d no es candidato de ninguna otra celda en la fila i .
- d no es candidato de ninguna otra celda en la columna j .
- d no es candidato de ninguna otra celda en la región que contiene a la celda (i,j)

Cuando esto se cumple podemos asegurar que la celda (i,j) debe ser ocupada por el dígito d .

En el sudoku de la figura, la posición **(1,5)** tiene como candidato exclusivo al **1**, puesto que es un candidato de esa celda y no es candidato de ninguna otra celda dentro de la misma fila.

Procedimiento General

Las reglas de *candidato único* y *candidato exclusivo* nos permiten deducir cuál es el número que debe ocupar una determinada celda. Aplicando sucesivamente estas reglas podemos llegar a resolver una cantidad importante de sudokus (aunque no todos).

El método general puede esbozarse de la siguiente manera:

1. Para cada celda libre (i,j) hacer lo siguiente:
 - Si existe d candidato único o exclusivo de (i,j) , ocupar la celda (i,j) con el número d .
2. Si el tablero está lleno, terminar.
3. Si el tablero no está lleno y se logró ocupar alguna celda en el último paso, volver al paso 1.
4. Si el tablero no está lleno y no se logró ocupar ninguna celda en el último paso, terminar.

Observaciones:

- Si terminamos por el paso 2, se resolvió el sudoku con éxito.
- Si terminamos por el paso 4, significa que el sudoku no es resoluble únicamente por reglas simples. Esto ocurre con sudokus de nivel *medio* y *difícil*.
- Cada vez que se ocupa una celda los conjuntos de candidatos de las celdas que están en la misma columna, fila y región sufren variantes. Por eso es necesario inspeccionar nuevamente la matriz desde el principio si hubo cambios.

Se pide

Escribir un programa pascal que implemente el procedimiento general que se describe más arriba. Este programa ingresa un tablero de sudoku desde la entrada estándar y despliega para cada paso, cada uno de los candidatos únicos o exclusivos que haya encontrado. Finalmente despliega el tablero con todas las celdas que logró completar.

En esta tarea sólo se pide implementar las técnicas que llamamos *candidato único* y *candidato exclusivo*. Ud. puede implementar otras técnicas además de las solicitadas, en caso de hacerlo, estas técnicas deben aplicarse luego de haber agotado la aplicación de las técnicas simples que solicitamos. La salida de su programa agregará al final de la salida solicitada el resultado de la aplicación de otras técnicas. El agregado de otras técnicas es opcional y no es requerido para aprobar la tarea. Existen muchos sitios que describen técnicas de resolución de sudoku, por ejemplo:

- <http://www.simes.clara.co.uk/programs/sudokutechniques.html>.
- <http://www.angusj.com/sudoku/hints.php>

La entrada del programa

La entrada se realizará de a una fila por línea. Las celdas no ocupadas se representan con un 0. Se supone que la entrada es correcta, y no importa que comportamiento adopte el programa en caso de que no lo sea. Esto significa que se puede suponer que siempre se ingresarán 9 líneas de 9 dígitos cada una y además que el sudoku correspondiente tiene solución.

Para el sudoku que aparece como ejemplo más arriba, la entrada será:

```
007604003
640509027
050000010
000170200
010948060
008065000
070000080
360801052
100407600
```

Cada línea consiste de 9 dígitos sin dejar ningún espacio. Luego del último dígito de una fila viene el fin de línea.

La salida del programa

La salida que produce el programa es la siguiente:

```
Recorrida: 1
(1,5) -> 1 (exclusivo)
(1,7) -> 5 (exclusivo)
(1,8) -> 9 (unico)
(2,3) -> 1 (exclusivo)
(2,5) -> 3 (exclusivo)
(2,7) -> 8 (unico)
(3,3) -> 3 (exclusivo)
```

```

(3,4) -> 7 (exclusivo)
(3,5) -> 8 (exclusivo)
(3,6) -> 2 (unico)
(3,7) -> 4 (unico)
(3,9) -> 6 (unico)
(4,3) -> 6 (exclusivo)
(4,6) -> 3 (unico)
(4,8) -> 4 (unico)
(4,9) -> 8 (exclusivo)
(5,7) -> 3 (exclusivo)
(5,9) -> 5 (unico)
(6,1) -> 4 (exclusivo)
(6,2) -> 3 (exclusivo)
(6,4) -> 2 (unico)
(6,8) -> 7 (unico)
(7,4) -> 3 (unico)
(7,6) -> 6 (unico)
(7,9) -> 4 (exclusivo)
(8,3) -> 4 (exclusivo)
(8,5) -> 9 (unico)
(8,7) -> 7 (unico)
(9,2) -> 8 (exclusivo)
(9,8) -> 3 (unico)
(9,9) -> 9 (unico)
Recorrida: 2
(1,1) -> 8 (exclusivo)
(1,2) -> 2 (unico)
(3,1) -> 9 (unico)
(4,1) -> 5 (unico)
(4,2) -> 9 (unico)
(5,1) -> 7 (exclusivo)
(5,3) -> 2 (unico)
(6,7) -> 9 (exclusivo)
(6,9) -> 1 (unico)
(7,1) -> 2 (unico)
(7,3) -> 9 (exclusivo)
(7,5) -> 5 (unico)
(7,7) -> 1 (unico)
(9,3) -> 5 (unico)
(9,5) -> 2 (unico)
8 2 7 6 1 4 5 9 3
6 4 1 5 3 9 8 2 7
9 5 3 7 8 2 4 1 6
5 9 6 1 7 3 2 4 8
7 1 2 9 4 8 3 6 5
4 3 8 2 6 5 9 7 1
2 7 9 3 5 6 1 8 4
3 6 4 8 9 1 7 5 2
1 8 5 4 2 7 6 3 9

```

Observaciones:

- Las celdas libres se van recorriendo de arriba hacia abajo y de izquierda a derecha.
- El orden en que se realiza la recorrida es relevante ya que al colocar un número en una celda se están descartando

candidatos para las celdas que restan recorrer.

- Si un número es a la vez candidato único y exclusivo se reportará como único.
- Si en el último paso no se ocupa ninguna celda se mostrará el mensaje: `No se encontro candidato unico o exclusivo`
En este caso, puede continuar su programa con la aplicación de otras técnicas de resolución. Para ello debe desplegar luego de la salida que se describe más arriba, algo así:

```
=== otras tecnicas ===
. . .
. . .
```

Su programa debe producir **exactamente la misma salida** que se describe más arriba y en cada uno de los ejemplos de sesiones. De lo contrario puede haber problemas con la corrección automática de la tarea.

En breve, dejaremos disponible un programa modelo. En caso de dudas sobre aspectos no contemplados en esta letra, se sugiere imitar el comportamiento del programa modelo.

Se puede utilizar

Esta tarea puede resolverse con los conceptos vistos en el curso hasta la semana del 24 de octubre inclusive. De todos modos no hay restricciones respecto a utilizar otros conceptos que se vean en las siguientes clases.

Se debe utilizar el lenguaje Pascal tal como fue dado en el curso. Éste es el Pascal estándar con algunos agregados a saber:

- Utilización de `else` en la instrucción `case`. Si no se especifica `else` se ejecutará la sentencia siguiente al `case` en caso de no coincidir con ninguna de las etiquetas.
- Evaluación por **circuito corto** de las operaciones booleanas (`and` y `or` condicional).

No está permitido utilizar facilidades de Turbo Pascal que no forman parte del estándar y no se dan en el curso. Así por ejemplo, no se pueden utilizar ninguna de las palabras siguientes: `uses`, `cr1scr`, `gotoxy`, `crt`, `readkey`, `shortint`, `longint`, `string`, etc.

En esta tarea como en todos los problemas de este curso, se valorará además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios impartidos en el curso. De esta manera, se hará énfasis en buenas prácticas de programación que lleven a un código legible, bien documentado y mantenible, tales como:

- indentación adecuada
- utilización correcta y apropiada de las estructuras de control,
- código claro y legible
- algoritmos razonablemente eficientes
- utilización de comentarios que documenten y complementen el código
- utilización de constantes simbólicas
- nombres mnemotécnicos para variables, constantes, etc.
- utilización de funciones y procedimientos

Ejemplos de sesiones

En el siguiente ejemplo no es posible resolver el sudoku con las reglas simples de candidato único y candidato exclusivo.

```
500200680
000067031
003800007
```

```

000720004
150000023
400053000
300006400
780490000
014005002
Recorrida: 1
(1,3) -> 1 (exclusivo)
(1,5) -> 3 (exclusivo)
(1,9) -> 9 (unico)
(2,2) -> 4 (exclusivo)
(2,4) -> 5 (exclusivo)
(2,7) -> 2 (unico)
(3,1) -> 2 (exclusivo)
(3,2) -> 6 (exclusivo)
(3,6) -> 9 (exclusivo)
(3,7) -> 5 (unico)
(3,8) -> 4 (unico)
(4,2) -> 3 (exclusivo)
(4,8) -> 5 (exclusivo)
(5,5) -> 4 (exclusivo)
(5,6) -> 8 (unico)
(7,4) -> 1 (unico)
(8,6) -> 2 (unico)
(8,7) -> 3 (exclusivo)
(8,8) -> 1 (exclusivo)
(9,4) -> 3 (unico)

```

```

Recorrida: 2
(1,2) -> 7 (unico)
(1,6) -> 4 (unico)
(3,5) -> 1 (unico)
(4,6) -> 1 (unico)
(6,7) -> 1 (exclusivo)

```

```

Recorrida: 3
No se encontro candidato unico o exclusivo

```

```

5 7 1 2 3 4 6 8 9
0 4 0 5 6 7 2 3 1
2 6 3 8 1 9 5 4 7
0 3 0 7 2 1 0 5 4
1 5 0 0 4 8 0 2 3
4 0 0 0 5 3 1 0 0
3 0 0 1 0 6 4 0 0
7 8 0 4 9 2 3 1 0
0 1 4 3 0 5 0 0 2
=== otras tecnicas ===
. . .

```

Otro ejemplo complejo no resoluble por métodos simples:

```

700302050
500000030
064005000

```

```
020804060
100000004
040701080
000500270
070000005
090206003
Recorrida: 1
(2,6) -> 7 (exclusivo)
(3,1) -> 3 (exclusivo)
(4,1) -> 9 (unico)
(5,2) -> 5 (exclusivo)
(5,3) -> 8 (exclusivo)
(5,7) -> 7 (exclusivo)
(6,1) -> 6 (unico)
(6,3) -> 3 (unico)
(7,2) -> 3 (exclusivo)
(8,1) -> 2 (exclusivo)
(9,3) -> 5 (exclusivo)
(9,5) -> 7 (exclusivo)
Recorrida: 2
(2,3) -> 2 (exclusivo)
(3,9) -> 7 (exclusivo)
(4,3) -> 7 (unico)
(4,7) -> 3 (exclusivo)
(4,9) -> 1 (unico)
(6,7) -> 5 (exclusivo)
(6,9) -> 2 (exclusivo)
Recorrida: 3
(1,3) -> 9 (exclusivo)
(3,8) -> 2 (exclusivo)
(4,5) -> 5 (unico)
(5,5) -> 2 (exclusivo)
(5,6) -> 3 (exclusivo)
(5,8) -> 9 (unico)
(6,5) -> 9 (unico)
(8,5) -> 3 (exclusivo)
Recorrida: 4
(5,4) -> 6 (unico)
Recorrida: 5
No se encontro candidato unico o exclusivo
7 0 9 3 0 2 0 5 0
5 0 2 0 0 7 0 3 0
3 6 4 0 0 5 0 2 7
9 2 7 8 5 4 3 6 1
1 5 8 6 2 3 7 9 4
6 4 3 7 9 1 5 8 2
0 3 0 5 0 0 2 7 0
2 7 0 0 3 0 0 0 5
0 9 5 2 7 6 0 0 3
=== otras tecnicas ===
. . .
```

La línea que anuncia la aplicación de otras técnicas no es necesaria si Ud. no implementa otras técnicas.

Recuperado desde <http://www.fing.edu.uy/inco/cursos/prog1/pm/field.php/Laboratorio/Tarea2>
Ultima modificación: 21 octubre de 2005, hora 14:27