The sudoku bundle for displaying, solving and generating Sudoku puzzles*

Peter Wilson†
The Herries Press

2006/02/19

Abstract

The sudoku bundle provides a coordinated set of packages for displaying, solving, and generating Sudoku puzzles. A set of over 50 puzzles is also supplied.

Contents

*This file (sudokubundle.dtx) has version number 1.0a, last revised on 2006/02/19.

†herries dot press at earthlink dot net

List of Figures

1 Introduction

In December 2005 the *PracTeX Journal* [PJ05] set a competition about Sudoku puzzles. Depending on their experience with TEX contestants were asked to (a) typeset a particular puzzle, (b) typeset a puzzle described in a 'sudoku' file, (c) create a solver for sudoku puzzles. I entered the competition with a printer and solver. Following from this it was no great effort to develop a matching Soduko puzzle generator.

The packages described here have been developed completely independently from Paul Abraham's sudoku package [Abr05], which I did not see until after I had finished work on them.

A Sudoku puzzle consists a 9 by 9 grid of cells with some of the cells containing a number between 1 and 9, such as is shown in Figure 1. The problem is to place a number between 1 and 9 in each cell such that no number appears more than once in each row and in each column and in each minor 3 by 3 grid. The solution to the example puzzle is shown in Figure 3. The puzzle and answer have been typeset using the printsudoku package.

Among many other sources the *Sudoku Solver by logic* website [SSBL] provides much information on Sudoku puzzles and their solutions, as does the *Sudoku Online* [SOL] website.

A Sudoku puzzle may be represented as a simple text file consisting of nine rows of numbers and dots, nine numbers and dots in each row. The numbers are the clues to the puzzle and the dots represent blanks in the grid. A sudoko file for the example puzzle is given in Figure 2.

This manual is typeset according to the conventions of the LATEX DOCSTRIP utility which enables the automatic extraction of the LATEX macro source files [MG04].

2 Usage

The bundle provides three packages; one for typesetting a puzzle, one for solving a puzzle, and the third for generating a puzzle. Each package requires the previous one(s).

A file is also provided that contains over 50 puzzles and their answers — all you have to do is run LaTeX on it, print it, and get to work on the puzzles (or give them to someone else).

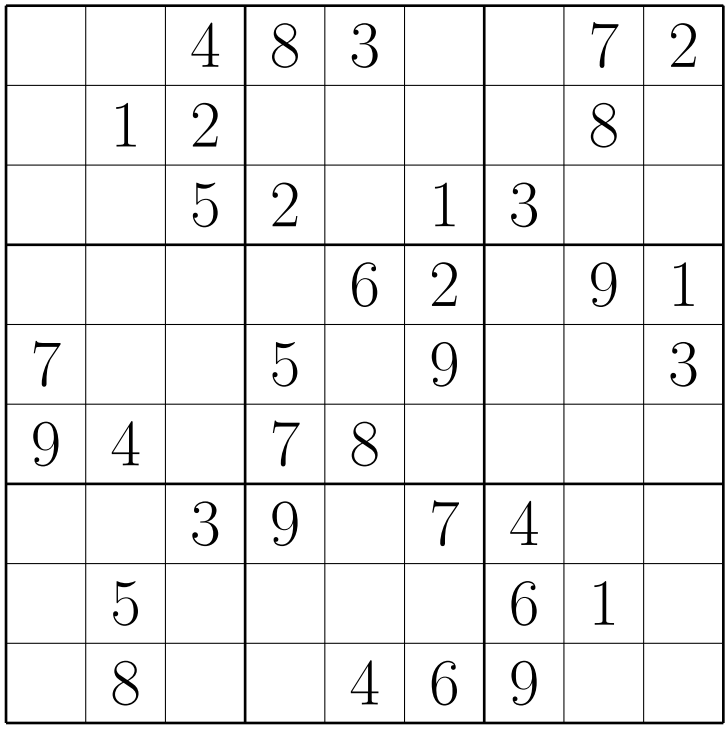

Figure 1: Example Sudoku puzzle

```
..483..72
.12....8.
..52.13..
....62.91
7..5.9..3
94.78....
..39.74..
.5....61.
.8..469..

Example puzzle
(anything can come after the nine puzzle lines)
```

Figure 2: A sudoku file for the example puzzle

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

Figure 3: Solution to example puzzle

2.1 Typesetting

The printsudoku package enables you to typeset a Sudoku puzzle given in a sudoku file.

`\sudoku` The command `\sudoku{`⟨*file*⟩`}` typesets the sudoku puzzle from the sudoku file named ⟨*file*⟩. For example:

```
\begin{center} \sudoku{puzzle1.sud} \end{center}
```

The command reads only the first nine lines in the file, which must contain the puzzle's description. Anything after these lines is ignored, so comments, such as a title or other explanatory text, can be put at the end of the file.

`\cluefont` Following `\cluefont{`⟨*font*⟩`}` the clues are typeset using the ⟨*font*⟩ font specification. The default is:

```
\cluefont{\Huge}
```

`\cellsize` The puzzle is typeset as a LaTeX `picture` and `\cellsize{`⟨*length*⟩`}` specifies the size of the cells in the grid. To match the `\cluefont` the default is:

```
\cellsize{2.5\baselineskip}
```

Figure 1 was set using the default `\cluefont` and `\cellsize`. On the other hand, Figure 3 was set via:

```
\begin{figure}
\centering
\cluefont{\normalsize}\cellsize{1.5\baselineskip}
\sudoku{examout.ans}
\caption{Solution to example puzzle}\label{fig:ans1}
\end{figure}
```

`\writepuzzle` `\puzzlefile` `\sudpuzznewline` The command `\writepuzzle`, which takes nine required arguments each of which is a line in the puzzle, writes the puzzle to the file `\puzzlefile`, where the default is:

```
\newcommand*{\puzzlefile}{puzz.sud}
```

The nine required arguments to `\writepuzzle` are the lines describing a puzzle as they would appear in a sudoku file. There is an optional tenth argument after the nine required arguments which you may use to write a comment at the end of the file. For instance, earlier I used the following to write out the example puzzle to the file `examout.sud`:

```
\renewcommand*{\puzzlefile}{examout.sud}
\writepuzzle%
{..483..72}%
{.12....8.}%
{..52.13..}%
{....62.91}%
{7..5.9..3}%
{94.78....}%
{..39.74..}%
{.5....61.}%
{.8..469..}%
[Example puzzle \sudpuzznewline
 (anything can come after the nine puzzle lines)]
```

And similarly for writing out the file with the solution.

If you want a multiline comment, use the `\sudpuzznewline` macro to start a new line (`\\` or `\newline` will not work).

2.2 Solving

The solvesudoku package lets you use LaTeX to try and solve a Sudoku puzzle.

`\sudokusolve` The macro `\sudokusolve{`⟨*file*⟩`}` attempts to find a solution to the puzzle in the sudoku file ⟨*file*⟩. It uses the printsudoku package for typesetting the puzzle.

It first prints the puzzle as given and then attempts to solve it. The solution, or as much as was obtainable, is typeset in a smaller font below the original. Here is the definition:

```
\newcommand*{\sudokusolve}[1]{%
  \begin{center}
  \sudoku{#1}
  \end{center}
  \getproblem{#1}
  \reduceallcells \keepreducing
  \writegame
  \noindent\begin{minipage}{\linewidth}
  THE ANSWER
  \begin{center}
  \cluefont{\normalsize}
  \cellsize{1.5\baselineskip}
  \sudoku{sud.out}
  \end{center}
  \end{minipage}}
```

You may use the macros in \sudokusolve separately if you wish.

\getproblem `\getproblem{⟨file⟩}` reads in the puzzle from the sudoku file ⟨*file*⟩ and initialises the solution.

\reduceallcells `\reduceallcells` takes a simple-minded approach to develop a solution to the puzzle.

\keepreducing If `\reduceallcells` does not completely solve the puzzle then `\keepreducing` applies increasingly sophisticated solution methods, which may or may not lead to a complete solution.

\writegame `\writegame` writes out a (partially) solved puzzle to the sudoku file `\sudsolnfile` (default `sud.out`). Later this is read in again by `\sudoku` to typeset the solution.

\sudsolnfile

Here's a small example of how you might use the package. This will keep asking on the terminal for sudoku files to solve.

```
%%% solvethem.tex   Solve Sudoku files
%%%                 Peter Wilson
 \documentclass{article}
 \usepackage{solvesudoku}
 \newcommand*{\solvefile}[1]{\begingroup
   \sudokusolve{#1}%
   \par
   \vspace{\baselineskip}
   Number of clues = \the\numcluesctr\ and difficulty = \the\difficultyctr.
 \endgroup}
 \def\yesans{y}
 \begin{document}
 \loop
   \typein[\getans]{New file? y/n}
   \ifx\yesans\getans
     \typein[\sudfile]{Enter the file name}
     \clearpage
     \begin{center} \Huge \sudfile \end{center}
     \solvefile{\sudfile}
 \repeat
 \end{document}
```

\numcluesctr These are two TEX `\count`s. `\numcluesctr` is the number of clues in the initial

\difficultyctr puzzle, and `\difficultyctr` is a numerical indication of how hard the solver had to work, but it doesn't really mean much as the solver has had successes with some puzzles with a high `\difficultyctr` rating but failed to solve some with medium values of `\difficultyctr`.

2.3 Generation

With the createsudoku package you can get LATEX to automatically generate a Sudoku puzzle. This package does require the use of Donald Arseneau's random code [Ars95] for generating random numbers.

\generategrid `\generategrid[⟨file⟩]` will generate a Sudoku puzzle starting with the solution

given in the sudoku file ⟨*file*⟩. If the file is not given then it will start from a solution provided by the package.

The puzzle is generated by taking the solution and randomly exchanging rows and columns for some (random) number of times. Then clues are eliminated until it is no longer soluble by `\sudokusolve`. The last clue that was deleted is put back and then the resulting grid is presented as the puzzle.

An initial set of clues are deleted before the solver is called. It can happen that the puzzle even then is too difficult for the solver, so you have to check the solution as presented is a complete one for the puzzle.

`\genfile` `\prevfile` `\currfile`

The generated puzzle, and its solution, is typeset and the puzzle is also output in the sudoku file `\genfile`. The default is:
`\newcommand*{\genfile}{gensud.sud}`
Two temporary sudoku files, `\prevfile` and `\currfile`, are used during the generation. The defaults are:
`\newcommand*{\prevfile}{genprev.sdx}`
`\newcommand*{\currfile}{gencurr.sdx}`
A pre-existing version of any of these three files will be overwritten.

`\setsudrandom`

The random number generater is initialised via a seed based on a number (the internal value of `\randomi`) in the range $-1 \leq$ `\randomi` ≤ 2147483646. If its value is -1 (set via `\setsudrandom{-1}`) then the actual seed will be calculated based on the time and date. For any given seed the sequence of the generated random numbers is the same. The value of the seed is output via a `\typeout` if you need to repeat a generation.

You should not attempt to set `\randomi` directly but use `\setsudrandom{`⟨*num*⟩`}`.

The default setting is:
`\setsudrandom{-1}`
and as that means that the seed depends on the time and date, a new puzzle will be generated each time (unless the time interval is very short). The maximum number of puzzles that can be generated from one initial solution is 2,147,483,646 which is the range of the random number generator.

`\initialelimination`

The macro `\initialelimination` eliminates an initial set of solutions from the randomized starting grid. Various predefined sets are provided and you can either provide your own definition for `\initialelimination` or `\let` it to one of the predefined sets, e.g.,
`\let\initialelimination\elimcrossandnines`

`\elimseventeen` `\elimnum` `\elimcross` `\elimex` `\elimcrossandnines` `\elimcrossandex` `\elimcrossandexandnines`

`\elimseventeen` randomly eliminates 17 solutions from the grid — apparently the solution to a puzzle can be ambiguous if two numbers are completely absent. `\elimnum{`⟨*num*⟩`}` eliminates every solution ⟨*num*⟩ (e.g. `\elimnum{7}` will eliminate every 7 from the grid). `\elimcross` eliminates all numbers from column 5 and from row 5, while `\elimex` eliminates all numbers on the diagonals.

`\elimcrossandnines` is a combination of `\elimcross` and `\elimnum{9}`. Similarly the macros `\elimcrossandex` and `\elimcrossandexandnines` are combinations of some of the earlier macros.

3 Code

3.1 The printsudoku package

```
⟨*print⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{printsudoku}[2006/02/16 v1.0 typeset sudoku puzzles]

```

`\s@dread` `\s@dwrite` We are going to be reading and writing some files.

```
\newread\s@dread
\newwrite\s@dwrite

```

`\reads@dgame` `\reads@dgame{`⟨*file*⟩`}` opens a sudoku file ⟨*file*⟩ for reading.

```
%%%% open game file
\newcommand*{\reads@dgame}[1]{%
  \closein\s@dread
  \openin\s@dread=#1}

```

`\halfs@dcell` `\cellsize` `\cluefont` We're going to use a `picture` environment for drawing the grid, and to keep numbers neatly we'll use the length `\halfs@dcell` as the `\unitlength`. `\cellsize{`⟨*length*⟩`}` is the use command for setting the size of a cell. The clues are typeset using the `\cluefont` font.

```
%%% set size of a game cell
\newlength\halfs@dcell
\newcommand*{\cellsize}[1]{%
  \halfs@dcell=#1\relax
  \halfs@dcell=0.5\halfs@dcell}
\cellsize{2.5\baselineskip}
%% set font for the numbers
\newcommand*{\cluefont}[1]{\def\s@dfont{#1}}
\cluefont{\Huge}

```

`\s@dncol` `\s@dnrow` `\s@dcolpos` `\s@drowpos` Counts for looping and cell positions

```
%% counts for looping and cell positions
\newcount\s@dncol
\newcount\s@dnrow
\newcount\s@dcolpos
\newcount\s@drowpos

```

`\adds@dclues` `\adds@dclues` inserts the clues into the game grid. It reads the sudoku file line by line insert the clues into the grid for one line before going on to the next line.

```
%% insert the clues into the game array
\newcommand*{\adds@dclues}{%
  \firsts@dcluetrue
```

```
  \s@dnrow\@ne
  \loop
    \ifnum\s@dnrow<10\relax
    \read\s@dread to \s@dline
    \firsts@dcluetrue
    \s@drowpos=\s@dnrow
    \multiply\s@drowpos\tw@
    \advance\s@drowpos\m@ne
    \s@dncol\@ne
    \dos@dcols
    \advance\s@dnrow\@ne
  \repeat
  \closein\s@dread}

```

\dos@dcols `\dos@dcols` inserts one row of clues into the game grid.

```
%% insert a row of clues
\newcommand*{\dos@dcols}{%
\bgroup
  \loop
    \ifnum\s@dncol<10\relax
    \s@dcolpos=\s@dncol
    \multiply\s@dcolpos\tw@
    \advance\s@dcolpos\m@ne
    \put(\s@dcolpos,-\s@drowpos){\makebox(0,0){\s@dfont\gets@dclue}}%
    \advance\s@dncol\@ne
  \repeat
\egroup}

```

\gettwo \nowt \istchar \restchars \splitoff `\splitoff{`⟨*string*⟩`}` gets the next character in a string. The next character is made available as `\istchar` and the remainder of the string as `\restchars`. This is explained in a forthcoming *Glisterings* column in *TUGboat*.

```
%% get the next character in a string
\def\gettwo#1#2\nowt{%
  \gdef\istchar{#1}\gdef\restchars{#2}}
\def\splitoff#1{\gettwo#1\nowt}

```

\s@dfstop We use this for checking if a character extracted from a string is a '.'.

```
%% a full stop (period)
\gdef\s@dfstop{.}

```

\iffirsts@dclue \firsts@dcluetrue \firsts@dcluefalse \gets@dclue `\gets@dclue` gets the next clue (character) from a line of clues. We do slightly different things if we are dealing with the first clue or the others.

```
%% get the next clue in the line
\newif\iffirsts@dclue
\firsts@dcluetrue
\newcommand*{\gets@dclue}{%
```

```
  \iffirsts@dclue
    \expandafter\splitoff\expandafter{\s@dline}%
    \global\firsts@dcluefalse
  \else
    \expandafter\splitoff\expandafter{\restchars}%
  \fi
  \ifx\s@dfstop\istchar
  \else
    \istchar
  \fi}

```

`\s@dgame` `\s@dgame` typesets the grid, then adds in the clues.

```
%% typeset the grid, then add the clues
\newcommand*{\s@dgame}{%
  \setlength\unitlength\halfs@dcell
  \begin{picture}(18,18)(0,-18)
    \thinlines
    \multiput(0,0)(2,0){10}{\line(0,-1){18}}
    \multiput(0,0)(0,-2){10}{\line(1,0){18}}
    \thicklines \linethickness{1pt}
    \multiput(0,0)(6,0){4}{\line(0,-1){18}}
    \multiput(0,0)(0,-6){4}{\line(1,0){18}}
    \adds@dclues
  \end{picture}}

```

`\sudoku` `\sudoku{`⟨*file*⟩`}` reads a game from ⟨*file*⟩ and then typesets it.

```
%% the whole shebang, where #1 is the name of the game file
\newcommand*{\sudoku}[1]{%
  \reads@dgame{#1}%
  \s@dgame}

```

`\puzzlefile` `\writepuzzle` takes nine arguments — the nine lines specifying a puzzle — and
`\writepuzzle` writes them to the `\puzzlefile` file.

`\writes@dpuzzend` `\writes@dpuzzend[`⟨*comment*⟩`]` writes ⟨*comment*⟩ at the end of the `\puzzlefile`
`\sudpuzznewline` file.

`\sudpuzznewline` provides a `\newline` macro for use in a comment being written to a puzzle file.

```
\newcommand*{\puzzlefile}{puzz.sud}
\newcommand*{\writepuzzle}[9]{%
  \immediate\closeout\s@dwrite
  \immediate\openout\s@dwrite=\puzzlefile
  \immediate\write\s@dwrite{#1}%
  \immediate\write\s@dwrite{#2}%
  \immediate\write\s@dwrite{#3}%
  \immediate\write\s@dwrite{#4}%
  \immediate\write\s@dwrite{#5}%
```

```
  \immediate\write\s@dwrite{#6}%
  \immediate\write\s@dwrite{#7}%
  \immediate\write\s@dwrite{#8}%
  \immediate\write\s@dwrite{#9}%
  \writes@dpuzzend}
\newcommand*{\writes@dpuzzend}[1][\@empty]{
  \ifx\@empty #1\else
    \immediate\write\s@dwrite{ }%
    \immediate\write\s@dwrite{#1}%
  \fi
  \immediate\closeout\s@dwrite}
\newcommand*{\sudpuzznewline}{^^J}
```

The end of the package.

```
⟨/print⟩
```

3.2 The solvesudoku package

The package requires the printsudoku package to do some puzzle typesetting.

```
⟨*solve⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{solvesudoku}[2006/02/16 v1.0 solve sudoku puzzles]
\RequirePackage{printsudoku}
```

3.2.1 Setup

\commentary It can be useful to control the amount of output to the log file.

```
\newcommand{\commentary}[1]{\typeout{#1}}
```

The following counts are defined in the printsudoku package:
`\s@dncol, \s@dnrow, \s@dcolpos, \s@drowpos`

\s@lcnta \solcnt \tenscnt \tempcntz \tempcnty \tmpsetctr \tmpsetansctr \boxctr \sumctr \toprangectr A bunch of \counts. They could probably be reduced in number but they help me to keep the code less mysterious.

```
%%% too many counts, maybe they should be reduced
\newcount\s@lcnta     % a cell number
\newcount\solcnt      % number of solved cells
\newcount\settonumcnt
\newcount\tenscnt
\newcount\tempcntz
\newcount\tempcnty
\newcount\tmpsetctr
\newcount\tmpsetansctr
\newcount\boxctr
\newcount\sumctr
\newcount\toprangectr
```

```
\lonecellctr
 \reducedctr 141 \newcount\lonecellctr
 \numlistctr 142 \newcount\reducedctr
\difficultyctr 143 \newcount\numlistctr
 \s@dtempcnta 144 \newcount\difficultyctr
 \s@dtempcntb 145 \newcount\s@dtempcnta
\s@dtemploopcnta 146 \newcount\s@dtempcntb
 \maxrangectr 147 \newcount\s@dtemploopcnta
     \asetctr 148 \newcount\maxrangectr
   \secondctr 149 \newcount\asetctr
              150 \newcount\secondctr

  \digitictr
  \digitiictr 151 \newcount\digitictr
\numdigitsctr 152 \newcount\digitiictr
 \numcluesctr 153 \newcount\numdigitsctr
              154 \newcount\numcluesctr
              155

\iffirsts@dclue
 \iffirstchar 156 %%% \iffirsts@dclue% defined in printsudoku package
\ifnotgotthechar 157 \newif\iffirstchar
\ifsetchanged 158 \newif\ifnotgotthechar
   \ifchanged 159 \newif\ifsetchanged
\ifanychanged 160 \newif\ifchanged
\iflonerchanged 161 \newif\ifanychange
\ifstilldigits 162 \newif\iflonerchanged
    \ifkeepon 163 \newif\ifstilldigits
\ifpairchanged 164 \newif\ifkeepon
              165 \newif\ifpairchanged
              166
```

`\newknt` `\useknt` `\createsudsets`

`\newknt{`⟨*name*⟩`}` creates a new count `\name`, which can include analphabetic characters. `\useknt{`⟨*name*⟩`}` calls the `\name` count. These are like the LaTeX kernel `\@namedef` and `\@nameuse`. In particular we will use 81 'numbered' counts, one for each cell in the sudoku grid, using `\createsudsets` to generate them.

```
167 %% create and use a numbered count
168 %% we'll use 81 of these, one for each cell in the grid
169 \newcommand*{\newknt}[1]{\expandafter\newcount\csname #1\endcsname}
170 \newcommand*{\useknt}[1]{\csname #1\endcsname}
171 %% create potential answer sets
172 \newcommand*{\createsudsets}{%
173 \global\s@lcnta\@ne
174 \global\solcnt\z@
175 \loop
176   \ifnum\s@lcnta<82\relax
177   \newknt{s@lans\the\s@lcnta}%
178   \advance\s@lcnta\@ne
179 \repeat}
```

```
\createsudsets
```

3.2.2 Binary solution sets

The major concern was deciding on the datastructure for the problem (the value of `\useknt{\s@lans\the\s@lcnta}` for the `\s@lcnta` cell). In the end I used a 9 digit 'binary solution set' for the potential solutions for each cell. ([111111111] ↔ 123456789 and [101010101] ↔ 13579). A cell solution, say N, is represented as the 'set' -N (e.g., a potential solution 3 is represented as [001000000] and the actual solution 3 is represented by [−3]).

`\initialisesuddata` This sets each of the 81 cells to 111111111.

```
%% initialise potential answer sets
\newcommand*{\initialisesuddata}{\bgroup
\global\s@lcnta\@ne
\global\solcnt\z@
\global\difficultyctr\z@
\global\numcluesctr\z@
\loop
  \ifnum\s@lcnta<82\relax
  \global\useknt{s@lans\the\s@lcnta}=111111111\relax
  \advance\s@lcnta\@ne
\repeat
\egroup}

%%%%%%%%%%%%%%%%%%%%%% utility macros for 'binary solution sets' %%%%%%%%%%%%%%%%
```

`\settonum` `\settonum{`⟨*set*⟩`}{`⟨*cnt*⟩`}` converts a potential solution binary set ⟨*set*⟩ to the corresponding set of digits; that is `[11....1] -> 12...9`. The result of the conversion is assigned to the `\count` ⟨*cnt*⟩ which must be supplied by the calling macro. If the set is negative then the result is that number (e.g.`[-3] -> -3`). If the set contains only a single non-zero entry, that is converted to the negative of the corresponding digit (e.g. `[100] -> -7`).

```
%%\settonum{110011...}{cnt}
%% converts a potential solution set to a number ([11....1] -> 12...9)
%% returns the input if it is negative ([-3] -> -3)
%% returns a negative number if the set represents a single digit ([10] -> -8)
\newcommand*{\settonum}[2]{%
  \settonumcnt=#1\relax
  \tempcnty=\z@
  \tenscnt=\@ne
  \ifnum\settonumcnt<\z@ % just return the number
    \tempcnty=\settonumcnt
    #2=\tempcnty
  \else
    \ifodd\settonumcnt    % 9 flagged
```

```
    \tempcntz=9\relax
    \multiply\tempcntz \tenscnt
    \advance \tempcnty by \tempcntz
    \multiply\tenscnt 10\relax
  \fi
  \divide\settonumcnt by 10\relax
  \ifodd\settonumcnt   % 8 flagged
    \tempcntz=8\relax
    \multiply\tempcntz \tenscnt
    \advance \tempcnty \tempcntz
    \multiply\tenscnt 10\relax
  \fi
  \divide\settonumcnt by 10\relax
  \ifodd\settonumcnt   % 7 flagged
    \tempcntz=7\relax
    \multiply\tempcntz \tenscnt
    \advance \tempcnty \tempcntz
    \multiply\tenscnt 10\relax
  \fi
  \divide\settonumcnt by 10\relax
  \ifodd\settonumcnt   % 6 flagged
    \tempcntz=6\relax
    \multiply\tempcntz \tenscnt
    \advance \tempcnty \tempcntz
    \multiply\tenscnt 10\relax
  \fi
  \divide\settonumcnt by 10\relax
  \ifodd\settonumcnt   % 5 flagged
    \tempcntz=5\relax
    \multiply\tempcntz \tenscnt
    \advance \tempcnty \tempcntz
    \multiply\tenscnt 10\relax
  \fi
  \divide\settonumcnt by 10\relax
  \ifodd\settonumcnt   % 4 flagged
    \tempcntz=4\relax
    \multiply\tempcntz \tenscnt
    \advance \tempcnty \tempcntz
    \multiply\tenscnt 10\relax
  \fi
  \divide\settonumcnt by 10\relax
  \ifodd\settonumcnt   % 3 flagged
    \tempcntz=3\relax
    \multiply\tempcntz \tenscnt
    \advance \tempcnty \tempcntz
    \multiply\tenscnt 10\relax
  \fi
  \divide\settonumcnt by 10\relax
  \ifodd\settonumcnt   % 2 flagged
    \tempcntz=2\relax
```

```
      \multiply\tempcntz \tenscnt
      \advance \tempcnty \tempcntz
      \multiply\tenscnt 10\relax
    \fi
    \divide\settonumcnt by 10\relax
    \ifodd\settonumcnt   % 1 flagged
      \tempcntz=1\relax
      \multiply\tempcntz \tenscnt
      \advance \tempcnty \tempcntz
%%      \multiply\tenscnt 10\relax
    \fi
    \ifnum \tempcnty<10
      \ifnum\tempcnty>\z@
        \tempcnty = -\tempcnty
      \fi
    \fi
    #2=\tempcnty
\fi}
```

`\numofnuminset` `\numofnuminset{`⟨*digit*⟩`}{`⟨*set*⟩`}{`⟨*cnt*⟩`}` set the `\count` ⟨*cnt*⟩to the number of times the digit ⟨*digit*⟩ is represented in the binary set ⟨*set*⟩. For example, the number of digits in the set `[200000013]` are `1 -> 2`, `2 -> 0`, ..., `8 -> 1` and `9 -> 3`.

```
%%\numofnuminset{digit}{set}{returnctr}
%%  returns the number of times digit is represented in the set
%% e.g. if the set is [200000013] then 1 -> 2, 2 -> 0,... 8 -> 1, 9 -> 3
\newcommand*{\numofnuminset}[3]{%
  \tmpsetctr=#2\relax
  \tmpsetansctr=\tmpsetctr
  \ifnum\tmpsetctr<\z@% solution, not a set
    \tmpsetansctr=\z@
  \else
    \ifcase #1\relax
    \or               % 1
      \divide\tmpsetansctr by 100000000\relax
    \or               % 2
      \divide\tmpsetansctr by 10000000\relax
      \tmpsetctr=\tmpsetansctr
      \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
      \advance\tmpsetansctr -\tmpsetctr
    \or               % 3
      \divide\tmpsetansctr by 1000000\relax
      \tmpsetctr=\tmpsetansctr
      \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
      \advance\tmpsetansctr -\tmpsetctr
    \or               % 4
      \divide\tmpsetansctr by 100000\relax
      \tmpsetctr=\tmpsetansctr
      \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
```

```
    \advance\tmpsetansctr -\tmpsetctr
  \or               % 5
    \divide\tmpsetansctr by 10000\relax
    \tmpsetctr=\tmpsetansctr
    \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
    \advance\tmpsetansctr -\tmpsetctr
  \or               % 6
    \divide\tmpsetansctr by 1000\relax
    \tmpsetctr=\tmpsetansctr
    \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
    \advance\tmpsetansctr -\tmpsetctr
  \or               % 7
    \divide\tmpsetansctr by 100\relax
    \tmpsetctr=\tmpsetansctr
    \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
    \advance\tmpsetansctr -\tmpsetctr
  \or               % 8
    \divide\tmpsetansctr by 10\relax
    \tmpsetctr=\tmpsetansctr
    \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
    \advance\tmpsetansctr -\tmpsetctr
  \or               % 9
    \tmpsetctr=\tmpsetansctr
    \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
    \advance\tmpsetansctr -\tmpsetctr
  \else             % error
    \tmpsetansctr=\z@
  \fi
 \fi
 #3=\tmpsetansctr}
```

`\deletenumfromset` `\deletenumfromset{`⟨*digit*⟩`}{`⟨*set*⟩`}{`⟨*cnt*⟩`}` removes the ⟨*digit*⟩ from the ⟨*set*⟩ and sets the `\count` ⟨*cnt*⟩ to the reulting modified set. If ⟨*digit*⟩ was the only member then the result is that digit negated. For example:
`9` from `[...11] -> [...10]`
`8` from `[0...010] -> -8`

```
%%\deletenumfromset{digit}{set}{returnctr}
%% deletes num from the [set], returning the modified set or a -ve number if
%% the digit was the only member.
%% e.g., 9 from [...11] -> [...10], 8 from [0...010] -> -8
\newcommand*{\deletenumfromset}[3]{
  \global\setchangedfalse
  \tmpsetctr=#2\relax
  \tmpsetansctr=#2\relax
\ifnum\tmpsetctr<\z@ % represents a solved number, do nothing
\else
  \ifcase #1\relax
  \or               % 1
```

```
348     \divide\tmpsetctr by 100000000\relax
349     \ifodd\tmpsetctr%               it's there
350       \advance\tmpsetansctr -100000000\relax
351       \global\setchangedtrue
352     \fi
353   \or               % 2
354     \divide\tmpsetctr by 10000000\relax
355     \ifodd\tmpsetctr%               it's there
356       \advance\tmpsetansctr -10000000\relax
357       \global\setchangedtrue
358     \fi
359   \or               % 3
360     \divide\tmpsetctr by 1000000\relax
361     \ifodd\tmpsetctr%               it's there
362       \advance\tmpsetansctr -1000000\relax
363       \global\setchangedtrue
364     \fi
365   \or               % 4
366     \divide\tmpsetctr by 100000\relax
367     \ifodd\tmpsetctr%               it's there
368       \advance\tmpsetansctr -100000\relax
369       \global\setchangedtrue
370     \fi
371   \or               % 5
372     \divide\tmpsetctr by 10000\relax
373     \ifodd\tmpsetctr%               it's there
374       \advance\tmpsetansctr -10000\relax
375       \global\setchangedtrue
376     \fi
377   \or               % 6
378     \divide\tmpsetctr by 1000\relax
379     \ifodd\tmpsetctr%               it's there
380       \advance\tmpsetansctr -1000\relax
381       \global\setchangedtrue
382     \fi
383   \or               % 7
384     \divide\tmpsetctr by 100\relax
385     \ifodd\tmpsetctr%               it's there
386       \advance\tmpsetansctr -100\relax
387       \global\setchangedtrue
388     \fi
389   \or               % 8
390     \divide\tmpsetctr by 10\relax
391     \ifodd\tmpsetctr%               it's there
392       \advance\tmpsetansctr -10\relax
393       \global\setchangedtrue
394     \fi
395   \or               % 9
396     \ifodd\tmpsetctr%               it's there
397       \advance\tmpsetansctr -1\relax
```

```
      \global\setchangedtrue
    \fi
  \fi
\fi
  #3=\tmpsetansctr}
```

`\displaystatus` This macro typesets a tabular showing the current real and potential solutions for the grid. A real solution is shown as a negative number; positive numbers are the potential solution digits for the cell.

```
%% typeset current answers plus potential answer digits
\newcommand*{\displaystatus}{%
\begin{tabular}{||c|c|c||c|c|c||c|c|c||}\hline\hline
\settonum{\the\useknt{s@lans1}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans2}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans3}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans4}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans5}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans6}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans7}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans8}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans9}}{\s@dtempcnta}\the\s@dtempcnta \\ \hline
\settonum{\the\useknt{s@lans10}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans11}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans12}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans13}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans14}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans15}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans16}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans17}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans18}}{\s@dtempcnta}\the\s@dtempcnta \\ \hline
\settonum{\the\useknt{s@lans19}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans20}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans21}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans22}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans23}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans24}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans25}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans26}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans27}}{\s@dtempcnta}\the\s@dtempcnta \\ \hline\hline
\settonum{\the\useknt{s@lans28}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans29}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans30}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans31}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans32}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans33}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans34}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans35}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans36}}{\s@dtempcnta}\the\s@dtempcnta \\ \hline
\settonum{\the\useknt{s@lans37}}{\s@dtempcnta}\the\s@dtempcnta &
```

```
\settonum{\the\useknt{s@lans38}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans39}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans40}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans41}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans42}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans43}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans44}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans45}}{\s@dtempcnta}\the\s@dtempcnta \\ \hline
\settonum{\the\useknt{s@lans46}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans47}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans48}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans49}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans50}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans51}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans52}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans53}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans54}}{\s@dtempcnta}\the\s@dtempcnta \\ \hline\hline
\settonum{\the\useknt{s@lans55}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans56}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans57}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans58}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans59}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans60}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans61}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans62}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans63}}{\s@dtempcnta}\the\s@dtempcnta \\ \hline
\settonum{\the\useknt{s@lans64}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans65}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans66}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans67}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans68}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans69}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans70}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans71}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans72}}{\s@dtempcnta}\the\s@dtempcnta \\ \hline
\settonum{\the\useknt{s@lans73}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans74}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans75}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans76}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans77}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans78}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans79}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans80}}{\s@dtempcnta}\the\s@dtempcnta &
\settonum{\the\useknt{s@lans81}}{\s@dtempcnta}\the\s@dtempcnta \\ \hline\hline
\end{tabular}}

```

3.2.3 Problem initialisation

```
%%%%%%%%%%%%%%%%%%% get the problem and initialise the solution %%%%%%%%%
```

491

\getproblem `\getproblem{`⟨*file*⟩`}` initialises the solution grid and reads the puzzle from the sudoku file ⟨*file*⟩. It calls `\initialsoln` to read the file and insert the clues into the grid.

```
%% read the problem and initialise the problem (solution)
\newcommand*{\getproblem}[1]{\bgroup
  \initialisesuddata
  \global\solcnt\z@
  \immediate\closein\s@dread
  \immediate\openin\s@dread=#1
  \initialsoln
  \typeout{Sudoku problem #1, \the\solcnt\space initial clues}
\egroup}
```

\initialsoln Get the clues from the puzzle file, line by line, and insert them into the grid. `\s@dline` holds one line from the file.

```
%% insert the clues from the input game file
\newcommand*{\initialsoln}{%
  \firsts@dcluetrue
  \s@dnrow\@ne
  \s@lcnta\@ne
  \loop
     \ifnum\s@dnrow<10\relax
     \read\s@dread to \s@dline
     \firsts@dcluetrue
     \s@dncol\@ne
     \dos@dinitialcols
     \advance\s@dnrow\@ne
  \repeat
  \immediate\closein\s@dread}
```

\dos@dinitialcols Insert the clues from one line of the puzzle file into one row of the grid.

```
%% insert the clues from one line of the game file
\newcommand*{\dos@dinitialcols}{\bgroup
  \loop
    \ifnum\s@dncol<10\relax
    \fixentry
    \advance\s@dncol\@ne
  \repeat
\egroup}
```

\fixentry Inserts a clue from the current line into the grid.

```
%% insert a clue from the current line of the game file
\newcommand*{\fixentry}{%
  \iffirsts@dclue
```

```
    \expandafter\splitoff\expandafter{\s@dline}%
    \global\firsts@dcluefalse
  \else
    \expandafter\splitoff\expandafter{\restchars}%
  \fi
  \ifx\s@dfstop\istchar
  \else
    \global\useknt{s@lans\the\s@lcnta}=-\istchar
    \global\advance\solcnt\@ne
    \global\advance\numcluesctr\@ne
  \fi
  \global\advance\s@lcnta\@ne}
```

3.3 Solution methods

Three facts are used to generate a solution.

1. A digit must be unique within a row, a column, and a block Thus, if a solution, say N, is known for a cell, then the potential solution N must be deleted from all cells in the row, column, and block that the solved cell is in. I have called this a simple reduction. The simple reductions are repeated until there are no changes to the potential solutions.

2. If among all the cells in a row (column, block) there is a digit that occurs only once among all the potential solutions, then that digit is the solution for its cell. I have termed this a loner. If a loner solution is found then simple reductions are repeated.

3. If among all the cells in a row (column, block) there are two digits which occur twice only in the row, each time as a pair (e.g., 39 and 39) then one or other of the two digits must be a solution for a cell in which the pairs occur. This means that the two digits in the pair cannot occur anywhere else in the row (column, block). If there are no loners then the problem is examined for pairs. If there is any change in a potential solution then the simple reduction process is applied.

The process stops either when all 81 cells have been solved or there is no change in any potential solution after going through all the above reductions.

There are more sophisticated solution techniques that could be applied but they are also increasingly difficult to code in TEX.

`\reduceacell` `\reduceacell` looks at the solution set for the current cell (the `\s@lcnta` cell) and if the cell has been solved then it deletes the solution digit from the row, column, and box in which the cell appears. This may result in new solutions for other cells.

Given a cell number N ($0 < N < 82$) and, using integer arithmetic, putting $M = (N - 1)/9$ then

- Cell number C at top of column: $C = N - 9M$
- Cell number R at left of row: $R = 9M + 1$
- Cell number b at top left of 3 by 3 box: $B = 27(M/3)+3((N-1-9M)/3)+1$

```
%% Eliminate a solution digit from its row, column and box.
%% This may result in some new solutions.
\newcommand*{\reduceacell}{\bgroup
  \ifnum\useknt{s@lans\the\s@lcnta}<\z@   % solved cell
    \s@dtempcnta=-\useknt{s@lans\the\s@lcnta}
%% get the col, row and box numbers
    \tempcnty=\s@lcnta       % M in the formulae
    \advance\tempcnty\m@ne   \divide\tempcnty 9\relax
    \tempcntz=\tempcnty      \multiply\tempcntz 9\relax
    \s@dcolpos=\s@lcnta         \advance\s@dcolpos -\tempcntz % col
    \s@drowpos\tempcntz         \advance\s@drowpos\@ne          % row
    \boxctr=\tempcnty        % box
    \divide\boxctr\thr@@     \multiply\boxctr 27\relax
    \tempcnty=\s@lcnta       % reuse \tempcnty
    \advance\tempcnty\m@ne   \advance\tempcnty -\tempcntz
    \divide\tempcnty\thr@@   \multiply\tempcnty \thr@@
    \advance\boxctr\tempcnty \advance\boxctr \@ne
%% reduce the row, column, and the box
    \reducerow{\the\s@drowpos}{\the\s@dtempcnta}
    \reducecol{\the\s@dcolpos}{\the\s@dtempcnta}
    \reducebox{\the\boxctr}{\the\s@dtempcnta}
  \fi
\egroup}

```

`\reducerow` `\reducerow{`⟨*row*⟩`}{`⟨*digit*⟩`}` eliminates ⟨*digit*⟩ from the binary solution sets of the cells in the row starting at cell number ⟨*row*⟩.

```
%% \reducerow{rowcellnumber}{digit}
%% simple reduction for digit in a row
\newcommand*{\reducerow}[2]{\bgroup
  \s@dncol=#1\relax
  \toprangectr=\s@dncol
  \advance\toprangectr 9\relax
  \bgroup
  \loop
    \ifnum\s@dncol<\toprangectr
    \deletenumfromset{#2}{\the\useknt{s@lans\the\s@dncol}}{\reducedctr}
    \ifsetchanged
      \global\changedtrue
      \settonum{\the\reducedctr}{\numlistctr}
      \ifnum\numlistctr<10\relax
        \global\advance\solcnt\@ne
\commentary{\the\solcnt\space solved
            (\the\s@dncol\space = \the\numlistctr\space in row starting #1)}%
```

```
        \global\useknt{s@lans\the\s@dncol}=\numlistctr
      \else
        \global\useknt{s@lans\the\s@dncol}=\reducedctr
      \fi
    \fi
    \advance\s@dncol\@ne
  \repeat
  \egroup
\egroup}
```

\reducecol `\reducecol{⟨col⟩}{⟨digit⟩}` eliminates ⟨*digit*⟩ from the binary solution sets of the cells in the column starting at cell number ⟨*col*⟩.

```
%% \reducecol{colcellnumber}{digit}
%% simple reduction of a digit in a column
\newcommand*{\reducecol}[2]{\bgroup
  \s@dnrow=#1\relax
  \toprangectr=\s@dnrow
  \advance\toprangectr 73\relax
  \bgroup
  \loop
    \ifnum\s@dnrow<\toprangectr
    \deletenumfromset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\reducedctr}
    \ifsetchanged
      \global\changedtrue
      \settonum{\the\reducedctr}{\numlistctr}
      \ifnum\numlistctr<10\relax
          \global\advance\solcnt\@ne
\commentary{\the\solcnt\space solved
            (\the\s@dnrow\space = \the\numlistctr\space in column starting #1)}%
          \global\useknt{s@lans\the\s@dnrow}=\numlistctr
      \else
        \global\useknt{s@lans\the\s@dnrow}=\reducedctr
      \fi
    \fi
    \advance\s@dnrow 9\relax
  \repeat
  \egroup
\egroup}
```

\reducebox `\reducebox{⟨box⟩}{⟨digit⟩}` eliminates ⟨*digit*⟩ from the binary solution sets of the cells in the box starting at cell number ⟨*box*⟩.

```
%% \reducebox{boxcellnumber}{digit}
%% simple reduction of a digit in a box
\newcommand*{\reducebox}[2]{\bgroup
  \s@dncol=#1\relax
  \toprangectr=\s@dncol
  \advance\toprangectr \thr@@
```

```
  \bgroup
  \loop
    \ifnum\s@dncol<\toprangectr
    \s@dnrow=\s@dncol
    \deletenumfromset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\reducedctr}
    \ifsetchanged
      \global\changedtrue
      \settonum{\the\reducedctr}{\numlistctr}
      \ifnum\numlistctr<10\relax
          \global\advance\solcnt\@ne
\commentary{\the\solcnt\space solved
            (\the\s@dnrow\space = \the\numlistctr\space in box starting #1)}%
          \global\useknt{s@lans\the\s@dnrow}=\numlistctr
      \else
        \global\useknt{s@lans\the\s@dnrow}=\reducedctr
      \fi
    \fi
    \advance\s@dnrow 9\relax
    \deletenumfromset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\reducedctr}
    \ifsetchanged
      \global\changedtrue
      \settonum{\the\reducedctr}{\numlistctr}
      \ifnum\numlistctr<10\relax
          \global\advance\solcnt\@ne
\commentary{\the\solcnt\space solved
            (\the\s@dnrow\space = \the\numlistctr\space in box starting #1)}%
          \global\useknt{s@lans\the\s@dnrow}=\numlistctr
      \else
        \global\useknt{s@lans\the\s@dnrow}=\reducedctr
      \fi
    \fi
    \advance\s@dnrow 9\relax
    \deletenumfromset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\reducedctr}
    \ifsetchanged
      \global\changedtrue
      \settonum{\the\reducedctr}{\numlistctr}
      \ifnum\numlistctr<10\relax
          \global\advance\solcnt\@ne
\commentary{\the\solcnt\space solved
            (\the\s@dnrow\space = \the\numlistctr\space in box starting #1)}%
          \global\useknt{s@lans\the\s@dnrow}=\numlistctr
      \else
        \global\useknt{s@lans\the\s@dnrow}=\reducedctr
      \fi
    \fi
    \advance\s@dncol \@ne
  \repeat
  \egroup
\egroup}

```

\reduceallcells `\reduceallcells` examines all 81 cells, eliminating solution digits from the binary solution sets of the other cells in the row, column, and box.

```
%% loop over every cell, removing solution digits from the potential sets
%% in the cell's row, column and box.
\newcommand*{\reduceallcells}{\bgroup
  \ifnum\difficultyctr<\@ne
    \global\difficultyctr\@ne
  \fi
  \s@lcnta\@ne
  \global\changedfalse
  \loop
    \ifnum\s@lcnta<82\relax
    \reduceacell
    \advance\s@lcnta\@ne
  \repeat
\egroup}

```

\keepreducing `\keepreducing` performs simple reductions on all the cells until either the puzzle is solved or no changes have been made. It then tries loner reductions, returning to simple reductions following any change. If no change has been made after loner reductions it then tries pair reductions. After any change it goes back to simple reductions. This iterative process continues until either the puzzle has been solved (`\solcnt` = 81) or no further changes can be made.

A measure of the difficulty of the puzzle could be made by incrementing a count each time a new kind of reduction is tried. The increment increases with the complexity of the reduction.

```
\newcommand*{\keepreducing}{%
  \anychangetrue
  \keepontrue
  \loop
    \ifkeepon
    \keepreducingcells
    \ifnum\solcnt<81\relax
%%%       \advance\difficultyctr 4\relax
      \reduceloners
    \fi
    \anychangefalse
    \iflonerchanged
      \anychangetrue
    \fi
    \ifanychange\else
      \ifnum\solcnt<81\relax
%%%         \advance\difficultyctr 8\relax
        \pairchangedfalse
        \reducepairs
        \ifpairchanged
          \anychangetrue
        \fi
```

```
      \fi
    \fi
    \checkkeepon
  \repeat
  \ifnum\solcnt<81\relax
    \global\multiply \difficultyctr 10\relax
  \fi
\typeout{************ Number of solutions = \the\solcnt}
\typeout{************ Difficulty rating = \the\difficultyctr}
}

```

`\checkkeepon` Determines whether or not to keep on trying to reach the solution.

```
\newcommand*{\checkkeepon}{%
  \ifnum\solcnt>80\relax
    \keeponfalse
  \fi
  \ifanychange\else
    \keeponfalse
  \fi}

```

`\keepreducingcells` Keep on trying the simple reduction.

```
%% Keep performing the simple reduction
\newcommand*{\keepreducingcells}{\bgroup
\commentary{Simple reductions}%
  \ifnum\difficultyctr<\tw@
    \global\difficultyctr\tw@
  \fi
  \changedtrue
  \loop
    \ifchanged
    \reduceallcells
    \checksimplereductions
  \repeat
\egroup}

```

`\checksimplereductions` Determines whether or not to keep on trying simple reductions.

```
\newcommand*{\checksimplereductions}{%
  \ifchanged\else
    \typesimplestatus
  \fi
  \ifnum\solcnt>80\relax%   solution found!
    \changedfalse
  \fi}

%%%%%%%%%%%%%%%%%%%% reduce loners %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

\reduceloners Reduce via the loner digits, trying rows first, and if no change then trying columns, and if still no change then trying boxes.

```
%% reduce the loner digits, per row, column and box
\newcommand*{\reduceloners}{%
\commentary{reduceloners}%
  \global\lonerchangedfalse
  \ifnum\difficultyctr<4\relax
    \global\difficultyctr=4\relax
  \fi
  \ifnum\solcnt>80\else
    \bgroup
    \reducerowloners
    \iflonerchanged\else
      \reducecolloners
      \iflonerchanged\else
        \reduceboxloners
      \fi
    \fi
    \typelonestatus
    \egroup
  \fi}

```

\reducerowloners Loop through the rows and reduce the first loner found.

```
%% Loop through the rows, reducing loners
\newcommand*{\reducerowloners}{%
  \sumctr\z@
  \s@dnrow\@ne
  \loop
    \ifnum\s@dnrow<10\relax
    \s@lcnta=\s@dnrow \advance\s@lcnta\m@ne
    \multiply\s@lcnta 9\relax \advance\s@lcnta\@ne
    \sumrowsets
    \ifnum\sumctr>\z@
      \getloner{\the\sumctr}{\s@dtempcntb}
      \ifnum\s@dtempcntb=\z@
      \else
        \reducelonerrowcell{\the\s@lcnta}{\the\s@dtempcntb}
        \iflonerchanged
          \advance\s@dnrow 10\relax
        \fi
      \fi
    \fi
    \advance\s@dnrow\@ne
  \repeat}

```

\sumrowsets Add together all the solution binary sets for the unsolved cells in the current row (\s@lcnta). The result is in \sumctr.

```
%% add the sets in the current row
\newcommand*{\sumrowsets}{%
  \bgroup
  \s@dncol=\s@lcnta
  \toprangectr=\s@dncol
  \advance\toprangectr 9\relax
  \global\sumctr\z@
  \loop
    \ifnum\s@dncol<\toprangectr
    \ifnum\useknt{s@lans\the\s@dncol}>\z@
      \global\advance\sumctr\useknt{s@lans\the\s@dncol}
    \fi
    \advance\s@dncol\@ne
  \repeat
  \egroup}

```

`\getloner` `\getloner{`⟨*setsum*⟩`}{`⟨*cnt*⟩`}` examines (the sum of) a solution set and returns the first loner (i.e., the first occurrence of the digit 1), or zero if there is none. The result is converted to a solution digit and assigned to the `\count` ⟨*cnt*⟩.

For example, given the set `[000200110]` it will return 7 (the set indicates two 4s, one 7, and one 8).

```
%% get the loner from a setsum
%% \getloner{setsum}{returnctr} returnes the (first) loner in a potential set,
%% or zero
\newcommand*{\getloner}[2]{\bgroup
  \ifnum #1>\z@
    \s@dtemploopcnta\@ne
    \loop
      \ifnum\s@dtemploopcnta<10\relax
      \numofnuminset{\the\s@dtemploopcnta}{#1}{\s@dtempcnta}
      \ifnum\s@dtempcnta=\@ne%         found a lone digit
        \s@dtempcnta=\s@dtemploopcnta
        \s@dtemploopcnta=100\relax
      \fi
      \advance\s@dtemploopcnta\@ne
    \repeat
    \ifnum\s@dtemploopcnta>99 \else \s@dtempcnta=\z@ \fi
    \global#2=\s@dtempcnta
  \else
    \s@dtempcnta=\z@
    \global#2=\s@dtempcnta
  \fi
\egroup}

```

`\reducelonerowcell` `\reducelonerowcell{`⟨*rowcell*⟩`}{`⟨*digit*⟩`}`. Given (the cell at) the start of a row and a lone digit, find the cell in the row with that digit in its solution set, and make the digit the solution of that cell.

```
%% \reducelonerowcell{startofrowcell}{digit}
%% given a row and a lone digit, find the cell containing the digit
%% and make the digit a solution of that cell.
\newcommand*{\reducelonerrowcell}[2]{\bgroup
  \s@dncol=#1
  \toprangectr=\s@dncol
  \advance\toprangectr 9\relax
  \ifnum #2<\@ne
    \advance\s@dncol 100\relax
  \fi
  \loop
    \ifnum\s@dncol<\toprangectr
    \numofnuminset{#2}{\the\useknt{s@lans\the\s@dncol}}{\lonecellctr}
    \ifnum\lonecellctr=\@ne%    this cell has the loner
      \lonecellctr=#2
      \global\useknt{s@lans\the\s@dncol}=-\lonecellctr
      \global\lonerchangedtrue
      \global\advance\solcnt\@ne
\commentary{\the\solcnt\space solved (#2 loner in row starting #1)}%
      \s@dncol=\toprangectr
    \fi
    \advance\s@dncol\@ne
  \repeat
\egroup}
```

`\reducecolloners` Loop through the columns and reduce the first loner found.

```
%% Loop through the columns, reducing loners
\newcommand*{\reducecolloners}{\bgroup
  \sumctr\z@
  \s@dncol\@ne
  \loop
    \ifnum\s@dncol<10\relax
    \s@lcnta=\s@dncol
    \sumcolsets
    \ifnum\sumctr>\z@
      \getloner{\the\sumctr}{\s@dtempcntb}
      \ifnum\s@dtempcntb=\z@
      \else
        \reducelonercolcell{\the\s@lcnta}{\the\s@dtempcntb}
        \iflonerchanged
          \advance\s@dncol 10\relax
        \fi
      \fi
    \fi
    \advance\s@dncol\@ne
  \repeat
\egroup}
```

`\sumcolsets` Sum all the solution sets of the unsolved cells in the current column (`\s@lcnta`). The result is in `\sumctr`.

```
%% add the sets in the current column
\newcommand*{\sumcolsets}{\bgroup
  \s@dnrow=\s@lcnta
  \toprangectr=\s@dnrow
  \advance\toprangectr 73\relax
  \global\sumctr\z@
  \loop
    \ifnum\s@dnrow<\toprangectr
    \ifnum\useknt{s@lans\the\s@dnrow}>\z@
      \global\advance\sumctr\useknt{s@lans\the\s@dnrow}
    \fi
    \advance\s@dnrow 9\relax
  \repeat
\egroup}

```

`\reducelonercolcell` `\reducelonercolcell{`⟨*colcell*⟩`}{`⟨*digit*⟩`}`. Given (the cell at) the start of a column and a lone digit, find the cell in the column with that digit in its solution set, and make the digit the solution of that cell.

```
%% \reducelonercolcell{startofcolcell}{digit}
%% given a col and a lone digit, find the cell containing the digit
%% and make the digit a solution of that cell.
\newcommand*{\reducelonercolcell}[2]{\bgroup
  \s@dnrow=#1
  \toprangectr=\s@dnrow
  \advance\toprangectr 73\relax
  \ifnum #2<\@ne
    \advance\s@dnrow 7300\relax
  \fi
  \loop
    \ifnum\s@dnrow<\toprangectr
    \numofnuminset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\lonecellctr}
    \ifnum\lonecellctr=\@ne%    this cell has the loner
      \lonecellctr=#2
      \global\useknt{s@lans\the\s@dnrow}=-\lonecellctr
      \global\lonerchangedtrue
      \global\advance\solcnt\@ne
\commentary{\the\solcnt\space solved (#2 loner in column starting #1)}%
      \s@dnrow=\toprangectr
    \fi
    \advance\s@dnrow 9\relax
  \repeat
  \egroup}

```

`\reduceboxloners` Loop through the boxes and reduce the first loner found.

```
%% Loop through the boxes, reducing loners
```

```
\newcommand*{\reduceboxloners}{\bgroup
  \sumctr\z@
  \boxctr\@ne
  \loop
    \ifnum\boxctr<10\relax
```

The next bit of code calculates the corner cell of the Nth box.

```
      \ifcase\boxctr
      \or \s@lcnta=\@ne
      \or \s@lcnta=4\relax
      \or \s@lcnta=7\relax
      \or \s@lcnta=28\relax
      \or \s@lcnta=31\relax
      \or \s@lcnta=34\relax
      \or \s@lcnta=55\relax
      \or \s@lcnta=58\relax
      \or \s@lcnta=61\relax
      \fi
      \sumboxsets
      \ifnum\sumctr>\z@
        \getloner{\the\sumctr}{\s@dtempcntb}
        \ifnum\s@dtempcntb=\z@
        \else
          \reducelonerboxcell{\the\s@lcnta}{\the\s@dtempcntb}
            \iflonerchanged
              \advance\boxctr 10\relax
            \fi
        \fi
      \fi
    \advance\boxctr\@ne
  \repeat
\egroup}

```

`\sumboxsets` Sum all the solution sets of the unsolved cells in the current box (`\s@lcnta`). The result is in `\sumctr`.

```
%% add the sets in the current box
\newcommand*{\sumboxsets}{\bgroup
  \s@dncol=\s@lcnta
  \toprangectr=\s@dncol
  \advance\toprangectr 3\relax
  \global\sumctr\z@
  \loop
    \ifnum\s@dncol<\toprangectr
    \s@dnrow=\s@dncol
    \ifnum\useknt{s@lans\the\s@dnrow}>\z@
      \global\advance\sumctr\useknt{s@lans\the\s@dnrow}
    \fi
    \advance\s@dnrow 9\relax
    \ifnum\useknt{s@lans\the\s@dnrow}>\z@
```

```
      \global\advance\sumctr\useknt{s@lans\the\s@dnrow}
    \fi
    \advance\s@dnrow 9\relax
    \ifnum\useknt{s@lans\the\s@dnrow}>\z@
      \global\advance\sumctr\useknt{s@lans\the\s@dnrow}
    \fi
    \advance\s@dncol\@ne
  \repeat
\egroup}
```

`\reducelonerboxcell` `\reducelonerboxcell{`⟨*boxcell*⟩`}{`⟨*digit*⟩`}`. Given (the cell at) the start of a box and a lone digit, find the cell in the box with that digit in its solution set, and make the digit the solution of that cell.

```
%% \reducelonerboxcell{startofboxcell}{digit}
%% given a box and a lone digit, find the cell containing the digit
%% and make the digit a solution of that cell.
\newcommand*{\reducelonerboxcell}[2]{\bgroup
  \s@dncol=\s@lcnta
  \toprangectr=\s@dncol
  \advance\toprangectr 3\relax
  \global\sumctr\z@
  \loop
    \ifnum\s@dncol<\toprangectr
    \s@dnrow=\s@dncol
    \numofnuminset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\lonecellctr}
    \ifnum\lonecellctr=\@ne%    this cell has the loner
      \lonecellctr=#2
      \global\useknt{s@lans\the\s@dnrow}=-\lonecellctr
      \global\lonerchangedtrue
      \global\advance\solcnt\@ne
\commentary{\the\solcnt\space solved (#2 loner in box starting #1)}%
      \s@dncol=\toprangectr
    \fi
    \iflonerchanged\else
      \advance\s@dnrow 9\relax
      \numofnuminset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\lonecellctr}
      \ifnum\lonecellctr=\@ne%    this cell has the loner
        \lonecellctr=#2
        \global\useknt{s@lans\the\s@dnrow}=-\lonecellctr
        \global\lonerchangedtrue
        \global\advance\solcnt\@ne
\commentary{\the\solcnt\space solved (#2 loner in box starting #1)}%
        \s@dncol=\toprangectr
      \fi
    \fi
    \iflonerchanged\else
      \advance\s@dnrow 9\relax
      \numofnuminset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\lonecellctr}
      \ifnum\lonecellctr=\@ne%    this cell has the loner
```

```
        \lonecellctr=#2
        \global\useknt{s@lans\the\s@dnrow}=-\lonecellctr
        \global\lonerchangedtrue
        \global\advance\solcnt\@ne
\commentary{\the\solcnt\space solved (#2 loner in box starting #1)}%
        \s@dncol=\toprangectr
      \fi
    \fi
    \advance\s@dncol\@ne
  \repeat
\egroup}

%%%%%%%%%%%%%%%%%%%% reduce pairs %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

`\reducepairs` Perform pair reduction, reducing the first pair found. First try by row, then by column, and lastly by box.

```
%% reduce the pairs, per row, column and box
\newcommand*{\reducepairs}{\bgroup
\commentary{reducepairs}% ************************************}%
  \global\pairchangedfalse
  \ifnum\difficultyctr<8\relax
    \global\difficultyctr=8\relax
  \fi
  \ifnum\solcnt>80\else
    \reducerowpairs
    \ifpairchanged\else
      \reducecolpairs
      \ifpairchanged\else
        \reduceboxpairs
      \fi
    \fi
  \fi
\egroup}
```

`\reducerowpairs` Loop through the rows and reduce the first pair found.

```
%% Loop through the rows, reducing pairs
\newcommand*{\reducerowpairs}{%
  \sumctr\z@
  \s@dnrow\@ne
  \loop
    \ifnum\s@dnrow<10\relax
    \s@lcnta=\s@dnrow \advance\s@lcnta\m@ne
    \multiply\s@lcnta 9\relax \advance\s@lcnta\@ne
    \sumrowsets
    \ifnum\sumctr>\z@
      \reducearowpair{\the\s@lcnta}
      \ifpairchanged
```

```
        \advance\s@dnrow 10\relax
      \fi
    \fi
    \advance\s@dnrow\@ne
  \repeat}
```

`\reducearowpair` `\reducearowpair{`⟨*rowcell*⟩`}` reduces a pair in the row starting with cell ⟨*rowcell*⟩.

```
%% reducearowpair{row}
\newcommand*{\reducearowpair}[1]{%
  \bgroup
  \s@dncol=#1\relax
  \toprangectr=\s@dncol
  \advance\toprangectr 9\relax
  \maxrangectr=\toprangectr
  \advance\maxrangectr\m@ne
  \global\sumctr\z@
  \loop
    \ifnum\s@dncol<\maxrangectr
    \asetctr=\useknt{s@lans\the\s@dncol}%
    \ifnum\asetctr>\z@
      \findrowpair
    \fi
    \advance\s@dncol\@ne
  \repeat
  \egroup}
```

`\findrowpair` `\findrowpair` attempts to find a pair in the current row.

```
\newcommand*{\findrowpair}{\bgroup
  \secondctr=\s@dncol
  \advance\secondctr\@ne
  \loop
    \ifnum\secondctr<\toprangectr
      \ifnum\asetctr=\useknt{s@lans\the\secondctr}
        \checksetforpair{\the\asetctr}
        \ifnum\numdigitsctr=\tw@
%% first row cell is \s@lcnta
%% first pair cell is \s@dncol, second pair cell is \secondctr
%% digits are \digitictr and \digitiictr
          \deleterowpairdigits
        \fi
      \fi
      \advance\secondctr\@ne
    \repeat
  \egroup}
```

`\checksetforpair` `\checksetforpair{`⟨*set*⟩`}` attempts to find a pair in a set. It sets `\digitictr` and `\digitiictr` as the first and second digits in the set and `\numdigitsctr` to

the total number of digits in the set (which is 2 for a pair).

```
%% \checksetforpair{set} returns with:
%%      \digitictr = first digit
%%      \digitiictr = second digit
%%      \numdigitsctr = number of digits (2 for a pair)
\newcommand*{\checksetforpair}[1]{\bgroup
  \s@dtemploopcnta\@ne
  \global\numdigitsctr\z@
  \global\digitictr\z@
  \global\digitiictr\z@
  \loop
    \ifnum\s@dtemploopcnta<10\relax
    \numofnuminset{\the\s@dtemploopcnta}{#1}{\s@dtempcnta}
    \ifnum\s@dtempcnta=\@ne
      \global\advance\numdigitsctr\@ne
      \ifcase\numdigitsctr
      \or \global\digitictr=\s@dtemploopcnta
      \or \global\digitiictr=\s@dtemploopcnta
      \else \advance\s@dtemploopcnta 10\relax
      \fi
    \fi
    \advance\s@dtemploopcnta\@ne
  \repeat
\egroup}

```

\deleterowpairdigits `\deleterowpairdigits` deletes the pair digits from non-pair cells in a row. The first cell in the row is `\s@lcnta`, the cells that contain the two pairs are `\s@dncol` and `\secondctr`, and the digits are `\digitictr` and `\digitiictr`.

```
%% first row cell is \s@lcnta
%% first pair cell is \s@dncol, second pair cell is \secondctr
%% digits are \digitictr and \digitiictr
\newcommand*{\deleterowpairdigits}{\bgroup
  \global\pairchangedfalse
  \s@dtemploopcnta=\s@lcnta
  \toprangectr=\s@dtemploopcnta \advance\toprangectr 9\relax
  \loop
    \ifnum\s@dtemploopcnta<\toprangectr
    \ifnum\s@dtemploopcnta=\s@dncol
    \else
      \ifnum\s@dtemploopcnta=\secondctr
      \else
        \deletenumfromset{\the\digitictr}%
                         {\the\useknt{s@lans\the\s@dtemploopcnta}}{\s@dtempcnta}
        \ifsetchanged
          \global\useknt{s@lans\the\s@dtemploopcnta}=\s@dtempcnta
          \global\pairchangedtrue
        \fi
        \deletenumfromset{\the\digitiictr}%
```

```
                        {\the\useknt{s@lans\the\s@dtemploopcnta}}{\s@dtempcnta}
      \ifsetchanged
        \global\useknt{s@lans\the\s@dtemploopcnta}=\s@dtempcnta
        \global\pairchangedtrue
      \fi
    \fi
  \fi
  \advance\s@dtemploopcnta\@ne
  \repeat
\egroup}
```

`\reducecolpairs` Loop through the rows and reduce the first pair found.

```
%% Loop through the columns, reducing pairs
\newcommand*{\reducecolpairs}{%
  \sumctr\z@
  \s@dncol\@ne
  \loop
    \ifnum\s@dncol<10\relax
    \s@lcnta=\s@dncol
    \sumcolsets
    \ifnum\sumctr>\z@
      \reduceacolpair{\the\s@lcnta}
      \ifpairchanged
        \advance\s@dncol 10\relax
      \fi
    \fi
    \advance\s@dncol\@ne
  \repeat}
```

`\reduceacolpair` `\reduceacolpair{`⟨*colcell*⟩`}` reduces a pair in the column starting with cell ⟨*colcell*⟩.

```
%% reduceacolpair{col}
\newcommand*{\reduceacolpair}[1]{\bgroup
  \s@dnrow=#1\relax
  \toprangectr=\s@dnrow
  \advance\toprangectr 73\relax
  \maxrangectr=\toprangectr
  \advance\maxrangectr -9\relax
  \global\sumctr\z@
  \loop
    \ifnum\s@dnrow<\maxrangectr
    \asetctr=\useknt{s@lans\the\s@dnrow}%
    \ifnum\asetctr>\z@
      \findcolpair
    \fi
    \advance\s@dnrow 9\relax
  \repeat
  \egroup}
```

```
1193
```

`\findcolpair` `\findcolpair` attempts to find a pair in the current column.

```
1194 \newcommand*{\findcolpair}{\bgroup
1195   \secondctr=\s@dnrow
1196   \advance\secondctr 9\relax
1197   \loop
1198     \ifnum\secondctr<\toprangectr
1199       \ifnum\asetctr=\useknt{s@lans\the\secondctr}
1200         \checksetforpair{\the\asetctr}
1201         \ifnum\numdigitsctr=\tw@
1202 %% first row cell is \s@lcnta
1203 %% first pair cell is \s@dnrow, second pair cell is \secondctr
1204 %% digits are \digitictr and \digitiictr
1205           \deletecolpairdigits
1206         \fi
1207       \fi
1208       \advance\secondctr 9\relax
1209     \repeat
1210   \egroup}
1211
```

`\deletecolpairdigits` `\deletecolpairdigits` deletes the pair digits from non-pair cells in a column. The first cell in the column is `\s@lcnta`, the cells that contain the two pairs are `\s@dncol` and `\secondctr`, and the digits are `\digitictr` and `\digitiictr`.

```
1212 %% first col cell is \s@lcnta
1213 %% first pair cell is \s@dnrow, second pair cell is \secondctr
1214 %% digits are \digitictr and \digitiictr
1215 \newcommand*{\deletecolpairdigits}{\bgroup
1216   \global\pairchangedfalse
1217   \s@dtemploopcnta=\s@lcnta
1218   \toprangectr=\s@dtemploopcnta \advance\toprangectr 73\relax
1219   \loop
1220     \ifnum\s@dtemploopcnta<\toprangectr
1221     \ifnum\s@dtemploopcnta=\s@dnrow
1222     \else
1223       \ifnum\s@dtemploopcnta=\secondctr
1224       \else
1225         \deletenumfromset{\the\digitictr}%
1226                          {\the\useknt{s@lans\the\s@dtemploopcnta}}{\s@dtempcnta}
1227         \ifsetchanged
1228           \global\useknt{s@lans\the\s@dtemploopcnta}=\s@dtempcnta
1229           \global\pairchangedtrue
1230         \fi
1231         \deletenumfromset{\the\digitiictr}%
1232                          {\the\useknt{s@lans\the\s@dtemploopcnta}}{\s@dtempcnta}
1233         \ifsetchanged
1234           \global\useknt{s@lans\the\s@dtemploopcnta}=\s@dtempcnta
1235           \global\pairchangedtrue
```

```
        \fi
      \fi
    \fi
  \advance\s@dtemploopcnta 9\relax
  \repeat
\egroup}
```

`\reduceboxpairs` Loop through the boxes and reduce the first pair found.

```
%% Loop through the boxes, reducing pairs
\newcommand*{\reduceboxpairs}{\bgroup
  \sumctr\z@
  \boxctr\@ne
  \loop
    \ifnum\boxctr<10\relax
      \ifcase\boxctr
      \or \s@lcnta=\@ne
      \or \s@lcnta=4\relax
      \or \s@lcnta=7\relax
      \or \s@lcnta=28\relax
      \or \s@lcnta=31\relax
      \or \s@lcnta=34\relax
      \or \s@lcnta=55\relax
      \or \s@lcnta=58\relax
      \or \s@lcnta=61\relax
      \fi
      \sumboxsets
      \ifnum\sumctr>\z@
        \reduceaboxpair{\the\s@lcnta}
        \ifpairchanged
           \advance\boxctr 10\relax
        \fi
      \fi
      \advance\boxctr\@ne
  \repeat
\egroup}
```

`\getnthboxcell` `\getnthboxcell{`⟨*n*⟩`}{`⟨*firstcell*⟩`}{`⟨*cnt*⟩`}`. Given the first cell (⟨*firstcell*⟩) in a box and a number, ⟨*n*⟩, between 1 and 9, set the `\count` ⟨*cnt*⟩ to the cell number of the ⟨*n*⟩th cell in the box.

```
%% \getnthboxcell{n}{firstcell}{returnctr}
\newcommand*{\getnthboxcell}[3]{\bgroup
  \s@dtempcnta=#2
  \ifcase#1
  \or
  \or \advance\s@dtempcnta 9\relax
  \or \advance\s@dtempcnta 18\relax
  \or \advance\s@dtempcnta \@ne
```

```
  \or \advance\s@dtempcnta 10\relax
  \or \advance\s@dtempcnta 19\relax
  \or \advance\s@dtempcnta \tw@
  \or \advance\s@dtempcnta 11\relax
  \or \advance\s@dtempcnta 20\relax
  \fi
  \global #3=\s@dtempcnta
\egroup}

```

\reduceaboxpair `\reduceaboxpair{`⟨*boxcell*⟩`}` reduces a pair in the box starting with cell ⟨*boxcell*⟩.

```
%% \reduceaboxpair{startofboxcell}
\newcommand*{\reduceaboxpair}[1]{\bgroup
  \s@dtemploopcnta\@ne
  \toprangectr=\s@dtemploopcnta
  \advance\toprangectr 9\relax
  \maxrangectr=\toprangectr
  \advance\maxrangectr\m@ne
  \loop
    \ifnum\s@dtemploopcnta<\maxrangectr
    \getnthboxcell{\the\s@dtemploopcnta}{#1}{\s@dtempcnta}
    \asetctr=\useknt{s@lans\the\s@dtempcnta}
    \ifnum\asetctr>\z@
      \s@dnrow=\s@dtemploopcnta
      \findboxpair{#1}
    \fi
    \advance\s@dtemploopcnta\@ne
  \repeat
\egroup}

```

\findboxpair `\findboxpair` attempts to find a pair in the current box, and then does a pair reduction for the box.

```
%% \findboxpair{startofboxcell}
\newcommand*{\findboxpair}[1]{%
  \bgroup
    \secondctr=\s@dtemploopcnta
    \advance\secondctr\@ne
    \loop
      \ifnum\secondctr<\toprangectr
      \getnthboxcell{\the\secondctr}{#1}{\s@dtempcntb}
      \ifnum\asetctr=\useknt{s@lans\the\s@dtempcntb}
        \checksetforpair{\the\asetctr}
        \ifnum\numdigitsctr=\tw@
%% first box cell is #1 (=\s@lcnta)
%% first pair cell is \s@dnrow, second pair cell is \secondctr
%% digits are \digitictr and \digitiictr
          \deleteboxpairdigits{#1}
        \fi
      \fi
```

```
      \advance\secondctr \@ne
    \repeat
  \egroup}
```

`\deleteboxpairdigits` `\deleteboxpairdigits{`⟨*boxcell*⟩`}` deletes the pair digits from non-pair cells in a box. The first cell in the box is ⟨*boxcell*⟩, the cells that contain the two pairs are `\s@dnrow` and `\secondctr`, and the digits are `\digitictr` and `\digitiictr`.

```
%% first box cell is #1
%% first pair cell is \s@dnrow, second pair cell is \secondctr
%% digits are \digitictr and \digitiictr
\newcommand*{\deleteboxpairdigits}[1]{\bgroup
  \global\pairchangedfalse
  \s@dtemploopcnta=\@ne
  \toprangectr=\s@dtemploopcnta \advance\toprangectr 9\relax
  \loop
    \ifnum\s@dtemploopcnta<\toprangectr
    \ifnum\s@dtemploopcnta=\s@dnrow
    \else
      \ifnum\s@dtemploopcnta=\secondctr
      \else
        \getnthboxcell{\the\s@dtemploopcnta}{#1}{\s@dtempcntb}
        \deletenumfromset{\the\digitictr}%
                         {\the\useknt{s@lans\the\s@dtempcntb}}{\s@dtempcnta}
        \ifsetchanged
          \global\useknt{s@lans\the\s@dtempcntb}=\s@dtempcnta
          \global\pairchangedtrue
        \fi
        \deletenumfromset{\the\digitiictr}%
                         {\the\useknt{s@lans\the\s@dtempcntb}}{\s@dtempcnta}
        \ifsetchanged
          \global\useknt{s@lans\the\s@dtempcntb}=\s@dtempcnta
          \global\pairchangedtrue
        \fi
      \fi
    \fi
  \advance\s@dtemploopcnta \@ne
  \repeat
\egroup}
```

3.3.1 Checking the solution

```
%%%%%%%%%%%%%%%%%%%%%%%%%%% check solution %%%%%%%%%%%%%%%%%%%%%%%%%%%
```

We have found a solution when all the binary sets are negative. It's less efficient, but probably more useful, to check each row, column, and block (3 times 81 tests instead of 81).

\checksolution Check the solution by rows, columns, and boxes.

```
\newcommand*{\checksolution}{%
  \checkrows
  \checkcols
  \checkboxes}
```

\checkrows Check the row solutions. A row is said to be correct if a solution has been assigned to each cell in the row, thus relying on the assumption that the solver never makes a mistake. The check is not sufficient for a solution obtained in another manner.

```
\newcommand*{\checkrows}{%
  \sumctr\z@
  \s@dnrow\@ne
  \loop
    \ifnum\s@dnrow<10\relax
    \s@lcnta=\s@dnrow \advance\s@lcnta\m@ne
    \multiply\s@lcnta 9\relax \advance\s@lcnta\@ne
    \sumrowsets
    \ifnum\sumctr=\z@\else
\typeout{Row \the\s@dnrow\space is incorrect}%
      Row \the\s@dnrow\space is incorrect. \\
    \fi
    \advance\s@dnrow\@ne
  \repeat}
```

\checkcols Check the column solutions.

```
\newcommand*{\checkcols}{%
\bgroup
  \sumctr\z@
  \s@dncol\@ne
  \loop
    \ifnum\s@dncol<10\relax
    \s@lcnta=\s@dncol
    \sumcolsets
    \ifnum\sumctr=\z@\else
\typeout{Column \the\s@dncol\space is incorrect}%
      Column \the\s@dncol\space is incorrect. \\
    \fi
    \advance\s@dncol\@ne
  \repeat
\egroup}
```

\checkboxes Check the box solutions.

```
\newcommand*{\checkboxes}{%
\bgroup
  \sumctr\z@
  \boxctr\@ne
```

```
  \loop
    \ifnum\boxctr<10\relax
\ifcase\boxctr
\or \s@lcnta=\@ne
\or \s@lcnta=4\relax
\or \s@lcnta=7\relax
\or \s@lcnta=28\relax
\or \s@lcnta=31\relax
\or \s@lcnta=34\relax
\or \s@lcnta=55\relax
\or \s@lcnta=58\relax
\or \s@lcnta=61\relax
\fi
    \sumboxsets
    \ifnum\sumctr=\z@\else
\typeout{Box \the\boxctr\space is incorrect}%
      Box \the\boxctr\space is incorrect. \\
    \fi
    \advance\boxctr\@ne
  \repeat
\egroup}

```

`\typesimplestatusX` May be used to display the cuurrent status of the (partial) solution following simple reductions.

```
%% typeset the current status of the partial solution
\newcommand*{\typesimplestatusX}{%
\par\noindent\begin{minipage}{\linewidth}
\begin{center}
Simple reductions stopped (with \the\solcnt\space solutions)\\
%\footnotesize
\displaystatus
\end{center}
\end{minipage}}

```

`\typelonestatusX` May be used to display the current status of the (partial) solution after a loner reduction.

```
%% typeset the current status after loner processing
\newcommand*{\typelonestatusX}{%
\par\noindent\begin{minipage}{\linewidth}
\begin{center}
Loners processed (with \the\solcnt\space solutions)\\
%\footnotesize
\displaystatus
\end{center}
\end{minipage}}

```

`\hideprogress` `\showprogess` Normally the progress towards a solution is not displayed. It will be displayed in

the document, though, after the \showprogress declaration. The \hideprogress declaration switches off the displays.

```
%% For curiosity as to how a solution is developing call the \showprogess
%% declaration (\hideprogess turns this off).
\newcommand*{\hideprogress}{%
  \let\typesimplestatus\relax
  \let\typelonestatus\relax}
\newcommand*{\showprogress}{%
  \let\typesimplestatus\typesimplestatusX
  \let\typelonestatus\typelonestatusX}
\hideprogress
%\showprogress

%%%%%%%%%%%%%%%%% write solution to a game file %%%%%%%%%%%%%%%
```

\sudsolnfile \writegame

\writegame writes the (partial) solution to the file \sudsolnfile (default sud.out).

```
%% write out the solution to game file \sudsolnfile.
\newcommand*{\sudsolnfile}{sud.out}
\newcommand*{\writegame}{%
  \immediate\closeout\s@dwrite
  \immediate\openout\s@dwrite=\sudsolnfile
  \s@dnrow\@ne
  \loop
    \ifnum\s@dnrow<81\relax
    \gatherline{\the\s@dnrow}
    \immediate\write\s@dwrite{\sudaline}
    \advance\s@dnrow 9\relax
  \repeat
  \immediate\closeout\s@dwrite}

```

\gatherline \sudaline

\gatherline{⟨*rowstartcell*⟩} collects the solutions from one row in the grid and converts them into the corresponding characters for one line in a sudoku file, putting the characters into the \sudaline macro.

```
%% \gatherline{colstart}
%% collect all the characters for one line in a game file
\newcommand*{\gatherline}[1]{\bgroup
  \gdef\sudaline{}
  \s@dncol=#1\relax
  \toprangectr=\s@dncol
  \advance\toprangectr 9\relax
  \loop
    \ifnum\s@dncol<\toprangectr
    \s@dtempcnta=\useknt{s@lans\the\s@dncol}
    \ifnum\s@dtempcnta>\m@ne
      \g@addto@macro{\sudaline}{.}
```

```
    \else
      \s@dtempcnta=-\s@dtempcnta
      \ifcase\s@dtempcnta
      \or \g@addto@macro{\sudaline}{1}
      \or \g@addto@macro{\sudaline}{2}
      \or \g@addto@macro{\sudaline}{3}
      \or \g@addto@macro{\sudaline}{4}
      \or \g@addto@macro{\sudaline}{5}
      \or \g@addto@macro{\sudaline}{6}
      \or \g@addto@macro{\sudaline}{7}
      \or \g@addto@macro{\sudaline}{8}
      \or \g@addto@macro{\sudaline}{9}
      \else \g@addto@macro{\sudaline}{.}
      \fi
    \fi
    \advance\s@dncol\@ne
  \repeat
\egroup}


%%%%%%%%%%%%%%%%%%%%%%%%%%%% THE END GOAL %%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%% \sudoksolve{<file>} %%%%%%%%%%%%%%%%%%%%%%%%%%
```

`\sudokusolve` `\sudokesolve{`⟨*file*⟩`}` reads a sudoku puzzle from the ⟨*file*⟩ sudoku file, typesets the problem, does its best to solve it. It writes the solution to the `\sudsolnfile` file, then reads it back in again and typesets it.

```
%% Read in and typeset the problem, solve it, and typeset the answer
\newcommand{\sudokusolve}[1]{%
  \begin{center}
  \sudoku{#1}%
  \end{center}
  \getproblem{#1}%
  \reduceallcells
  \keepreducing
  \writegame
  \noindent\begin{minipage}{\linewidth}
  THE ANSWER
  \begin{center}
  \cluefont\normalsize
  \cellsize{1.5\baselineskip}
  \sudoku{\sudsolnfile}
  \end{center}
  \end{minipage}}
```

The end of the package.

```
⟨/solve⟩
```

3.4 The createsudoku package

The createsudoku package lets you automatically generate Sudoku puzzles. The package requires Donald Arseneau's `random.tex` and also the solvesudoku package, which in turn requires the printsudoku package.

```
⟨*create⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{createsudoku}[2006/02/16 v1.0 create sudoku puzzles]
\input{random}
\RequirePackage{solvesudoku}

```

`\oldcommentary` `\commentary` `\gencommentary`

To avoid clutter it's a good idea to switch off solvesudoku's commentary. On the other hand it could be useful for this package to provide its own commentary.

```
\let\oldcommentary\commentary
\let\commentary\@gobble
\newcommand*{\gencommentary}[1]{\typeout{#1}}

```

`\genfile` `\prevfile` `\currfile`

We need a file to write out the puxzzle to, and we also need two temporary files.

```
\newcommand*{\genfile}{gensud.sud}
\newcommand*{\prevfile}{genprev.sdx}
\newcommand*{\currfile}{gencurr.sdx}

```

`\randomi` `\setsudrandom`

This sets the seed for the random number generator. The default is setting it to -1, which makes the seed depend on the date and time.

```
\randomi\m@ne
\newcommand*{\setsudrandom}[1]{%
  \randomi=#1\relax
  \ifnum\randomi<\@ne\else
    \nextrandom \nextrandom \nextrandom
    \typeout{random initialised to #1}%
  \fi}

```

`\generategrid` `\printsudresults`

`\generategrid[`⟨*file*⟩`]` generates a new puzzle. If the sudoko file ⟨*file*⟩ is specified this is taken for the starting grid, otherwise a default one is used. The starting grid should be a complete solution.

The pairs of rows, and pairs of columns, in the starting grid are exchanged in a random fashion (`\swaps`) and then the cell solutions are eliminated (`\elimclues`) until the puzzle is sufficiently complicated.

```
\newcommand*{\generategrid}[1][\@empty]{%
  \ifx\@empty #1
    \writestartgrid
    \getproblem{\prevfile}
  \else
    \getproblem{#1}
```

```
  \fi
  \swaps
  \elimclues
  \printsudresults}
\newcommand*{\printsudresults}{%
  \sudokusolve{\genfile}%
  }
```

`\writestartgrid` Write out an initial fully solved grid to `\prevfile`.

```
%% the starting data
\newcommand*{\writestartgrid}{%
  \immediate\closeout\s@dwrite
  \immediate\openout\s@dwrite=\prevfile
  \immediate\write\s@dwrite{123456789}
  \immediate\write\s@dwrite{456789123}
  \immediate\write\s@dwrite{789123456}
  \immediate\write\s@dwrite{234567891}
  \immediate\write\s@dwrite{567891234}
  \immediate\write\s@dwrite{891234567}
  \immediate\write\s@dwrite{345678912}
  \immediate\write\s@dwrite{678912345}
  \immediate\write\s@dwrite{912345678}
  \immediate\closeout\s@dwrite}
```

`\swaps` `\swaps` randomly exchanges the contents of pairs of columns and pairs of rows. In order to preserve the number uniqueness constraint for boxes, only pairs of columns/rows that are in the same box may be exchanged.

```
\newcommand*{\swaps}{\bgroup
  \setrannum{\maxrangectr}{36}{48}%
\gencommentary{number of swaps (36--48) = maxrangectr = \the\maxrangectr}
  \s@dtemploopcnta\@ne
  \loop
    \ifnum\s@dtemploopcnta<\maxrangectr
    \setrannum{\s@dtempcntb}{1}{18}%
%%\typeout{\space\space\space swap no \the\s@dtemploopcnta\space is \the\s@dtempcntb}
    \ifcase\s@dtempcntb
    \or \swapcolpair{1}{2}%
    \or \swapcolpair{1}{3}%
    \or \swapcolpair{2}{3}%
    \or \swapcolpair{4}{5}%
    \or \swapcolpair{4}{6}%
    \or \swapcolpair{5}{6}%
    \or \swapcolpair{7}{8}%
    \or \swapcolpair{7}{9}%
    \or \swapcolpair{8}{9}%
    \or \swaprowpair{1}{10}%
    \or \swaprowpair{1}{19}%
```

```
    \or \swaprowpair{10}{19}%
    \or \swaprowpair{28}{37}%
    \or \swaprowpair{28}{46}%
    \or \swaprowpair{37}{46}%
    \or \swaprowpair{55}{64}%
    \or \swaprowpair{55}{73}%
    \or \swaprowpair{64}{73}%
    \fi
    \advance\s@dtemploopcnta\@ne
  \repeat
\egroup}

```

`\swaprowpair` `\swaprowpair{`⟨*rowstartcell1*⟩`}{`⟨*rowstartcell2*⟩`}` swaps the contents of the two rows starting at ⟨*rowstartcell1*⟩ and ⟨*rowstartcell2*⟩.

```
%% swap the data in rows starting at #1, and #2
\newcommand*{\swaprowpair}[2]{\bgroup
  \s@dncol=#1\relax
  \s@dcolpos=#2\relax
  \s@dtemploopcnta\@ne
  \maxrangectr=10\relax
  \loop
    \ifnum\s@dtemploopcnta<\maxrangectr
    \s@dtempcnta=\useknt{s@lans\the\s@dncol}%
    \global\useknt{s@lans\the\s@dncol}=\useknt{s@lans\the\s@dcolpos}%
    \global\useknt{s@lans\the\s@dcolpos}=\s@dtempcnta
    \advance\s@dncol\@ne
    \advance\s@dcolpos\@ne
    \advance\s@dtemploopcnta\@ne
  \repeat
\egroup}

```

`\swapcolpair` `\swapcolpair{`⟨*colstartcell1*⟩`}{`⟨*colowstartcell2*⟩`}` swaps the contents of the two columns starting at ⟨*colstartcell1*⟩ and ⟨*colstartcell2*⟩.

```
%% swap the data in cols starting #1 and #2
\newcommand*{\swapcolpair}[2]{\bgroup
  \s@dnrow=#1\relax
  \s@drowpos=#2\relax
  \s@dtemploopcnta\@ne
  \maxrangectr=10\relax
  \loop
    \ifnum\s@dtemploopcnta<\maxrangectr
    \s@dtempcnta=\useknt{s@lans\the\s@dnrow}%
    \global\useknt{s@lans\the\s@dnrow}=\useknt{s@lans\the\s@drowpos}%
    \global\useknt{s@lans\the\s@drowpos}=\s@dtempcnta
    \advance\s@dnrow 9\relax
    \advance\s@drowpos 9\relax
    \advance\s@dtemploopcnta\@ne
```

```
  \repeat
\egroup}
```

`\elimclues` `\elimclues` first eliminates a set of clues from the (full) grid, writing the resulting puzzle to the two temporary files `\prevfile` and `\currfile`. It then eliminates clues one-by-one until the puzzle has been finally generated, the result having been written to the `\prevfile`, then writing it out to the file `\genfile`.

```
\newcommand*{\elimclues}{\bgroup
  \initialelimination
  \writestate{\prevfile}%
  \writestate{\currfile}%
  \elimcluesonebyone
  \getproblem{\prevfile}
  \writestate{\genfile}
\egroup}
```

`\elimseventeen` `\elimseventeen` randomly eliminates the solutions for 17 cells in the grid (apparently if two numbers are completely eliminated as clues in a puzzle, then at best the solution is ambiguous).

```
\newcommand*{\elimseventeen}{\bgroup
  \s@dtemploopcnta\@ne
  \maxrangectr 18\relax
  \loop
    \ifnum\s@dtemploopcnta<\maxrangectr
    \setrannum{\s@dtempcnta}{1}{81}
    \global\useknt{s@lans\the\s@dtempcnta}=111111111\relax
    \advance\s@dtemploopcnta\@ne
  \repeat
\egroup}
```

`\elimcross` Eliminate all solutions in column 5 and row 5.

```
%% eliminates all clues in column 5 and row 5
\newcommand*{\elimcross}{\bgroup
  \s@dtemploopcnta 5\relax
  \loop
    \ifnum\s@dtemploopcnta<78\relax%    78 = 5 + 73
    \global\useknt{s@lans\the\s@dtemploopcnta}=111111111\relax
    \advance\s@dtemploopcnta 9\relax
  \repeat
  \s@dtemploopcnta 37\relax
  \loop
    \ifnum\s@dtemploopcnta<46\relax%    46 = 37 + 9
    \global\useknt{s@lans\the\s@dtemploopcnta}=111111111\relax
    \advance\s@dtemploopcnta \@ne
  \repeat
\egroup}
```

```
1676
```

`\elimnum` `\elimnum{⟨num⟩}` (`0 < num < 10`) eliminates all instances of the ⟨*num*⟩ solution from the grid.

```
1677 \newcommand*{\elimnum}[1]{\bgroup
1678   \s@dtemploopcnta\@ne
1679   \loop
1680     \ifnum\s@dtemploopcnta<82\relax
1681     \ifnum\useknt{s@lans\the\s@dtemploopcnta}=-#1\relax
1682       \global\useknt{s@lans\the\s@dtemploopcnta}=111111111\relax
1683     \fi
1684     \advance\s@dtemploopcnta \@ne
1685   \repeat
1686 \egroup}
1687
```

`\elimcrossandnines` Eliminate all 9's and the solutions in column 5 and row 5.

```
1688 \newcommand*{\elimcrossandnines}{\bgroup
1689   \elimcross
1690   \s@dtemploopcnta\@ne
1691   \loop
1692     \ifnum\s@dtemploopcnta<82\relax
1693     \ifnum\useknt{s@lans\the\s@dtemploopcnta}=-9\relax
1694       \global\useknt{s@lans\the\s@dtemploopcnta}=111111111\relax
1695     \fi
1696     \advance\s@dtemploopcnta \@ne
1697   \repeat
1698 \egroup}
1699
```

`\elimex` Eliminate all solutions on the diagonals of the grid.

```
1700 \newcommand*{\elimex}{\bgroup
1701   \s@dtemploopcnta\@ne
1702   \loop
1703     \ifnum\s@dtemploopcnta<82\relax
1704     \global\useknt{s@lans\the\s@dtemploopcnta}=111111111\relax
1705     \advance\s@dtemploopcnta 10\relax
1706   \repeat
1707   \s@dtemploopcnta=9\relax
1708   \loop
1709     \ifnum\s@dtemploopcnta<81\relax
1710     \global\useknt{s@lans\the\s@dtemploopcnta}=111111111\relax
1711     \advance\s@dtemploopcnta 8\relax
1712   \repeat
1713 \egroup}
1714
```

`\elimcrossandex` Eliminate all solutions in column 5 and row 5 and on the diagonals.

```
1715 \newcommand*{\elimcrossandex}{\bgroup
```

```
  \elimcross
  \elimex
\egroup}
```

`\elimcrossandexandnines` Eliminate all solutions in column 5 and row 5 and on the diagonals, and all 9's.

```
\newcommand*{\elimcrossandexandnines}{\bgroup
  \elimcross
  \elimex
  \elimnum{9}
\egroup}
```

`\initialelimination` `\initialelimination` eliminates an initial set of solutions from the grid. Several predefined sets are provided.

```
\let\initialelimination\elimseventeen
\let\initialelimination\elimcross
\let\initialelimination\elimcrossandnines
\let\initialelimination\elimcrossandex
\let\initialelimination\elimcrossandexandnines
```

`\toomanyloops` `\elimcluesonebyone` Eliminate solutions from the grid one-by-one until the puzzle cannot be solved. Then reinstate the last solution deleted to produce the final puzzle. I'm not sure if this is needed, but just in case the process stops after 81 attempts to remove solutions (highly unlikely but the random number generator could keep producing numbers corresponding to clueless cells).

We use the `\currfile` and `\prevfile` to hold the current state of affairs and the previous state.

```
\newcount\toomanyloops
\newcommand*{\elimcluesonebyone}{\bgroup
%%%\typeout{***elimcluesonebyone}
\toomanyloops\z@
  \s@dtempcntb=81\relax
  \loop
    \ifnum\s@dtempcntb>80\relax
\advance\toomanyloops\@ne
```

Read the current puzzle from `\currfile` and try to solve it.

```
    \getproblem{\currfile}
    \trysolution
    \ifnum\solcnt=81\relax     % can delete another clue
```

The puzzle was solvable so try deleting another clue. Read the (unsolved) puzzle again from `\currfile` and write its state to `\prevfile`.

```
      \getproblem{\currfile}
      \writestate{\prevfile}
```

Now delete a clue, writing the resultant puzzle to `\currfile`, and go round the loop again.

```
      \deleteaclue
      \writestate{\currfile}%
      \s@dtempcntb=81\relax
    \else % done
```

The puzzle was not solvable, and the state before the last clue deletion is in file \prevfile.

```
      \typeout{Solver failed, solcnt = \the\solcnt}
      \s@dtempcntb=\@ne
    \fi
    \ifnum\toomanyloops>80\relax
      \typeout{Too much looping}
      \s@dtempcntb=\@ne
    \fi
  \repeat
\egroup}
```

\ifacluenotdeleted
\deleteaclue

Keep calling the random number generator until it returns a cell that has a solution, then delete the solution.

```
\newif\ifacluenotdeleted
\newcommand*{\deleteaclue}{\bgroup
  \acluenotdeletedtrue
  \loop
    \ifacluenotdeleted
    \setrannum{\s@dtempcnta}{1}{81}%
      \ifnum\useknt{s@lans\the\s@dtempcnta}<\z@%  found a clue
        \gencommentary{Deleting clue for cell \the\s@dtempcnta}%
        \global\useknt{s@lans\the\s@dtempcnta}=111111111\relax
        \acluenotdeletedfalse
      \fi
  \repeat
\egroup}
```

\trysolution

\trysolution attempts to solve the current puzzle state.

```
\newcommand*{\trysolution}{\bgroup
  \reduceallcells
  \keepreducing
\egroup}
```

\writestate

\writestate{⟨*file*⟩} writes the current state (clues and blanks) to the sudoku file ⟨*file*⟩.

```
%% \writestate{<outputfile>}
\newcommand*{\writestate}[1]{\bgroup
  \immediate\closeout\s@dwrite
  \immediate\openout\s@dwrite=#1
  \s@dnrow\@ne
  \loop
```

```
    \ifnum\s@dnrow<81\relax
    \gatherline{\the\s@dnrow}%
    \immediate\write\s@dwrite{\sudaline}%
    \advance\s@dnrow 9\relax
  \repeat
  \immediate\closeout\s@dwrite
\egroup}

```

The end of the package.

⟨/create⟩

4 Example sudoku puzzles

LaTeXing the following file will provide you with a selection of Sudoku puzzles. You can try solving them yourself and/or get Latex to do it for you. The solvesudoku package is able to solve most, but not all of the puzzles.

⟨∗examples⟩

```
\documentclass[twoside]{article}
\usepackage{comment}
\usepackage{url}
\usepackage{printsudoku}
\newcommand*{\Lpack}[1]{\textsf{#1}}
\title{Example Sudoku Puzzles}
\author{Peter Wilson}
\date{16 February 2006}
\raggedbottom
\begin{document}
\maketitle

    A Sudoku puzzle consists of a 9 by 9 array of squares, some of which have
numbers in them, the numbers ranging from 1 through 9. The problem is to
fill in all the squares such that each row, column, and 3 by 3 box includes
the numbers 1 through 9 with no repetitions.

    This is a collection of some 54 Sudoku puzzles which I have culled
from various places. The sources include, among others:
the \emph{Sudoku Solver}~\cite{SSBL} website (labelled SE, SM, SH and SVH),
\emph{The Daily Telegraph}~\cite{DT2} (labelled TG, TM, and TT),
\emph{The Seattle Times} (labelled ST),
and some automatically generated by the \Lpack{createsudoku}
package~\cite{SUDOKUBUNDLE} (labelled CS).

You can try solving them yourself and/or get
LaTeX to do it for
you. The \Lpack{solvesudoku} package~\cite{SUDOKUBUNDLE} is able to solve
most, but not all of the puzzles. Other LaTeX packages may do better or worse.

```

```
    The puzzles are presented in approximately increasing order of difficulty.
Each source, though, has its own set of categories which don't necessarily
agree with each other (and sometimes not within one source). I have included
an indication of the difficulty as stated by the relevant source.

The parameters for the puzzles created via \Lpack{createsudoku} are given in
the table.

\begin{table}[hbp]
\centering
\begin{tabular}{lll} \hline
Number & Seed & \verb?initialelimination? \\ \hline
CS1    & 1505263462 & \verb?\elimcrossandexandnines? \\
CS2    & 1495929734 & \verb?\elimcrossandex? \\
CS3    & 1487762722 & \verb?\elimcrossandexandnines? \\
CS4    & 1508763610 & \verb?\elimcrossandex? \\
CS5    & 839068723  & \verb?\elimseventeen? \\
CS6    & 833235143  & \verb?\elimcrossandexandnines? \\ \hline
\end{tabular}
\end{table}

\cleardoublepage

\cluefont{\LARGE}
\cellsize{2.25\baselineskip}

\markboth{EASY}{EASY}
\pagestyle{headings}

\renewcommand*{\puzzlefile}{se5.sud}
\writepuzzle%
{.2...5863}{56.2.3.9.}{.3...7251}%
{..975....}{..6..47.9}{.7..286..}%
{6.58...7.}{8....1..6}{3.7.6..4.}%
[SSBL easy 5]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SE5 (easy) \\
\sudoku{se5.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{se10.sud}
\writepuzzle%
{..6.497..}{.82.1.6..}{79..8.145}%
{649.5.27.}{..7.6..5.}{.3..72.96}%
{.2....81.}{.7..28...}{......5.7}%
[SSBL easy 10]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SE10 (easy) \\
```

```
\sudoku{se10.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{se15.sud}
\writepuzzle%
{4.67.39.2}{..16.84..}{.7...4..1}%
{5...4.21.}{.2.....6.}{.68.7...9}%
{6..5...9.}{..24.93..}{9.53.17.4}%
[SSBL easy 15]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SE15 (easy) \\
\sudoku{se15.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{se20.sud}
\writepuzzle%
{36.9..5..}{..54.8.2.}{.....78.9}%
{2....3.68}{.83.9....}{.4.7..2.3}%
{6.71..3..}{13.8.59.2}{..9.7..56}%
[SSBL easy 20]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SE20 (easy) \\
\sudoku{se20.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{se25.sud}
\writepuzzle%
{....63.5.}{6.5.1.839}{..1...2..}%
{823...7..}{.5...7...}{.64.9...1}%
{4.25796.3}{.9.1.65.8}{.1...89.7}%
[SSBL easy 25]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SE25 (easy) \\
\sudoku{se25.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{se30.sud}
\writepuzzle%
{.3.12....}{1.87..4.6}{..98....1}%
{5....7.3.}{.73.9.2.4}{.92..65..}%
{.4...8.5.}{3...7...8}{2..4.596.}%
[SSBL easy 30]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SE30 (easy) \\
\sudoku{se30.sud}
\end{center}\end{minipage}
```

```
1924
1925 \renewcommand*{\puzzlefile}{tg5.sud}
1926 \writepuzzle%
1927 {.6.1.9.8.}{.19...74.}{2.......1}%
1928 {..79.26..}{.3.....2.}{..14.35..}%
1929 {1.......5}{.75...23.}{.9.5.6.7.}%
1930 [DT2 gentle 5]
1931 \vfill
1932 \begin{minipage}{0.95\linewidth}\begin{center}
1933 TG5 (gentle) \\
1934 \sudoku{tg5.sud}
1935 \end{center}\end{minipage}
1936
1937 \renewcommand*{\puzzlefile}{tg10.sud}
1938 \writepuzzle%
1939 {....96..4}{..1.....2}{56...8...}%
1940 {2.8....9.}{9.63.52.7}{.3....4.6}%
1941 {...9...58}{7.....9..}{8..45....}%
1942 [DT2 gentle 10]
1943 \vfill
1944 \begin{minipage}{0.95\linewidth}\begin{center}
1945 TG10 (gentle) \\
1946 \sudoku{tg10.sud}
1947 \end{center}\end{minipage}
1948
1949 \renewcommand*{\puzzlefile}{tg15.sud}
1950 \writepuzzle%
1951 {...1.8...}{.57...18.}{98.....26}%
1952 {..67419..}{.........}{..42536..}%
1953 {72.....94}{.19...73.}{...3.9...}%
1954 [DT2 gentle 15]
1955 \vfill
1956 \begin{minipage}{0.95\linewidth}\begin{center}
1957 TG15 (gentle) \\
1958 \sudoku{tg15.sud}
1959 \end{center}\end{minipage}
1960
1961 \renewcommand*{\puzzlefile}{tg20.sud}
1962 \writepuzzle%
1963 {...7.9...}{.8.3.6.5.}{7.9...6.8}%
1964 {..42.18..}{.6.....4.}{..35.41..}%
1965 {9.2...5.7}{.4.8.7.1.}{...1.2...}%
1966 [DT2 gentle 20]
1967 \vfill
1968 \begin{minipage}{0.95\linewidth}\begin{center}
1969 TG20 (gentle) \\
1970 \sudoku{tg20.sud}
1971 \end{center}\end{minipage}
1972
1973 \renewcommand*{\puzzlefile}{tg25.sud}
```

```
\writepuzzle%
{7..4..5..}{.61..5.7.}{.9.7.1..6}%
{8...27...}{.........}{...34...1}%
{1..8.2.3.}{.7.1..62.}{..3..9..7}%
[DT2 gentle 25]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TG25 (gentle) \\
\sudoku{tg25.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tg30.sud}
\writepuzzle%
{37.4.....}{9....61.4}{....23...}%
{.4.1..9..}{.12...79.}{..5..7.4.}%
{...36....}{4.32....6}{.....9.57}%
[DT2 gentle 30]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TG30 (gentle) \\
\sudoku{tg30.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st1226.sud}
\writepuzzle%
{9....4.31}{7452.....}{.8..674..}%
{.26.4...5}{5..6.9..8}{4...2.37.}%
{..759..6.}{.....1823}{21.3....7}%
[Seattle Times 2005/12/26 1 star]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
ST1226 (1 star) \\
\sudoku{st1226.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st0206.sud}
\writepuzzle%
{17.42...5}{..2.1..96}{849..3...}%
{....718..}{.25...67.}{..826....}%
{...5..749}{53..9.1..}{6...48.23}%
[Seattle Times 2005/02/06 1 star]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
ST0206 (1 star) \\
\sudoku{st0206.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st0213.sud}
\writepuzzle%
{.94.2...7}{.36...1.5}{28.35....}%
```

```
{..92...5.}{6.21.49.8}{.7...63..}%
{....13.46}{1.5...78.}{8...4.29.}%
[Seattle Times 2006/02/13 1 star]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
ST0213 (1 star) \\
\sudoku{st0213.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{cs1.sud}
\writepuzzle%
{.46..731.}{2.36.4...}{87.3.1.45}%
{657...423}{.........}{.81...756}%
{43.8.6..1}{7..2..5.4}{..25.386.}%
[CreateSudoku 1 (level 2)]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
CS1 (level 2) \\
\sudoku{cs1.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{cs2.sud}
\writepuzzle%
{.7.3.265.}{...6..9.7}{64.9.8.2.}%
{..9...765}{.........}{756...4.2}%
{29.5...76}{...2.15.3}{..4..7.1.}%
[CreateSudoku 2 (level 2)]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
CS2 (level 2) \\
\sudoku{cs2.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{cs3.sud}
\writepuzzle%
{.5.8.71..}{..35.....}{78.2.1.56}%
{.34...8.1}{.........}{8.1...56.}%
{.4.7.6.12}{...4.36.8}{.781..34.}%
[CreateSudoku 3 (level 2)]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
CS3 (level 2) \\
\sudoku{cs3.sud}
\end{center}\end{minipage}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% moderate

\clearpage
\markboth{MODERATE}{MODERATE}

```

```
\renewcommand*{\puzzlefile}{sm5.sud}
\writepuzzle%
{..76..24.}{.46....9.}{92..85...}%
{8....29..}{..3...4..}{..49....6}%
{...73..24}{.5....86.}{.32..67..}%
[SSBL medium 5]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SM5 (medium) \\
\sudoku{sm5.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sm10.sud}
\writepuzzle%
{..57.82..}{4.......6}{.7..6..8.}%
{52.9.4.31}{.........}{86.1.7.54}%
{.9..3..6.}{7.......5}{..24.61..}%
[SSBL medium 10]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SM10 (medium) \\
\sudoku{sm10.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sm15.sud}
\writepuzzle%
{85...4..2}{4...2.9..}{...8...7.}%
{..6.....1}{.4..8..3.}{2.....5..}%
{.34..7...}{.87.9...6}{6..1...47}%
[SSBL medium 15]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SM15 (medium) \\
\sudoku{sm15.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sm20.sud}
\writepuzzle%
{..34.2.97}{..4....2.}{.7.6....8}%
{....9....}{.38...47.}{....1....}%
{4....5.1.}{.5....6..}{72.3.19..}%
[SSBL medium 20]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SM20 (medium) \\
\sudoku{sm20.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sm25.sud}
\writepuzzle%
```

```
{...4.79..}{7.6...2..}{..5..83.6}%
{56..298..}{..8..1...}{.3.7....5}%
{.......38}{....93...}{.4.5.....}%
[SSBL medium 25]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SM25 (medium) \\
\sudoku{sm25.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sm30.sud}
\writepuzzle%
{..58.....}{....79...}{..461..7.}%
{.5.3....4}{.23...65.}{9....8.3.}%
{.3..572..}{...42....}{.....61..}%
[SSBL medium 30]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SM30 (medium) \\
\sudoku{sm30.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tm35.sud}
\writepuzzle%
{5..3.8..6}{..4.5.8..}{2.8...1.5}%
{..62.74..}{.........}{..94.32..}%
{9.5...6.4}{..2.1.9..}{1..8.9..2}%
[DT2 moderate 35]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TM35 (moderate) \\
\sudoku{tm35.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tm40.sud}
\writepuzzle%
{...1...92}{9...7..36}{3....2...}%
{.3....7.9}{.8.9.5.6.}{4.5....8.}%
{...8....5}{54..2...7}{12...3...}%
[DT2 moderate 40]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TM40 (moderate) \\
\sudoku{tm40.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tm45.sud}
\writepuzzle%
{.1.2.8.3.}{..87.94..}{9.......6}%
{..9.8.5..}{6.......3}{..3.5.6..}%
```

```
{2.......7}{..73.51..}{.6.4.7.9.}%
[DT2 moderate 45]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TM45 (moderate) \\
\sudoku{tm45.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tm50.sud}
\writepuzzle%
{...9.2..4}{4..3.5...}{3.....7.6}%
{6..8..3..}{1.3...9.7}{..9..7..2}%
{9.5.....8}{...2.9..1}{2..7.4...}%
[DT2 moderate 50]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TM50 (moderate) \\
\sudoku{tm50.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tm55.sud}
\writepuzzle%
{7........}{24.1.8...}{.6..72..3}%
{..4.29.8.}{.7.....6.}{.9.46.7..}%
{1..68..3.}{...3.7.91}{........6}%
[DT2 moderate 55]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TM55 (moderate) \\
\sudoku{tm55.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tm60.sud}
\writepuzzle%
{724.9.5..}{.5..2...8}{.8.5.7...}%
{...7..64.}{....8....}{.79..6...}%
{...8.5.2.}{6...4..8.}{..8.3.174}%
[DT2 moderate 60]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TM60 (moderate) \\
\sudoku{tm60.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st0914.sud}
\writepuzzle%
{.6.29..8.}{..8....3.}{.1..78...}%
{.217.9..8}{6.......3}{7..6.492.}%
{...12..4.}{.7....3..}{.5..86.7.}%
[Seattle Times 2005/09/14  2 stars]
```

```
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
ST0914 (2 stars) \\
\sudoku{st0914.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st0207.sud}
\writepuzzle%
{.......46}{5..8.9.73}{...6..5.1}%
{.592..1..}{6...1...7}{..3..785.}%
{2.5..6...}{19.7.5..8}{34.......}%
[Seattle Times 2005/02/07 2 stars]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
ST0207 (2 stars) \\
\sudoku{st0207.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{cs4.sud}
\writepuzzle%
{...4.597.}{..67.83.2}{78.1.2.45}%
{891...75.}{.........}{567...423}%
{.1...4.67}{..89.15..}{.4....29.}%
[CreateSudoku 4 (level 8)]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
CS4 (level 4) \\
\sudoku{cs4.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{cs5.sud}
\writepuzzle%
{...6..978}{..498...2}{.97.21...}%
{..84.27..}{.7.....23}{3...6...9}%
{7.6..95..}{...87...1}{.2..4.867}%
[CreateSudoku 5 (level 4)]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
CS5 (level 4) \\
\sudoku{cs5.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{cs6.sud}
\writepuzzle%
{.126.4.7.}{6.5..73.2}{.7.3.1.4.}%
{756...423}{.........}{18......6}%
{.3.8.6..1}{2.1.....7}{..72...3.}%
[CreateSudoku 6 (level 4)]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
```

```
CS6 (level 4) \\
\sudoku{cs6.sud}
\end{center}\end{minipage}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% hard

\clearpage
\markboth{HARD}{HARD}

\renewcommand*{\puzzlefile}{sh5.sud}
\writepuzzle%
{..9..5...}{47...2...}{.836.4.1.}%
{.15....8.}{...3.7...}{.2....74.}%
{.4.5.187.}{...7...52}{...2..1..}%
[SSBL hard 5]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SH5 (hard) \\
\sudoku{sh5.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sh10.sud}
\writepuzzle%
{.3.921..5}{15.83....}{.........}%
{59......4}{37.....81}{6......53}%
{.........}{....19.26}{9..574.1.}%
[SSBL hard 10]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SH10 (hard) \\
\sudoku{sh10.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sh15.sud}
\writepuzzle%
{2..9.....}{.7..6..59}{...4...61}%
{7.1..2..4}{....4....}{8..1..9.6}%
{12...3...}{53..2..7.}{.....9..2}%
[SSBL hard 15]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SH15 (hard) \\
\sudoku{sh15.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sh20.sud}
\writepuzzle%
{.6.2...3.}{9...3...1}{....9.2..}%
{1..7.....}{.32...46.}{.....4..2}%
{..7.4....}{2...5...8}{.4...6.5.}%
```

```
[SSBL hard 20]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SH20 (hard) \\
\sudoku{sh20.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sh25.sud}
\writepuzzle%
{4....1..8}{..7.3845.}{..24...1.}%
{28.9.....}{..4...5..}{.....4.86}%
{.4...92..}{.2134.8..}{9..6....1}%
[SSBL hard 25]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SH25 (hard) \\
\sudoku{sh25.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sh30.sud}
\writepuzzle%
{....5..9.}{.53..78..}{78.4.....}%
{3....9.2.}{.72...38.}{.1.3....4}%
{.....1.35}{..58..41.}{.2..3....}%
[SSBL hard 30]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
SH30 (hard) \\
\sudoku{sh30.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tt100.sud}
\writepuzzle%
{...896..7}{..23.5...}{..8...5.1}%
{15...9...}{3.......9}{...6...54}%
{8.1...7..}{...9.18..}{6..428...}%
[DT2 tough 100]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TT100 (tough) \\
\sudoku{tt100.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tt105.sud}
\writepuzzle%
{..5.7..46}{6..25....}{.3...6...}%
{2..7.....}{.78...35.}{.....1..7}%
{...8...2.}{....97..5}{81..2.9..}%
[DT2 tough 105]
\vfill
```

```
\begin{minipage}{0.95\linewidth}\begin{center}
TT105 (tough) \\
\sudoku{tt105.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tt110.sud}
\writepuzzle%
{.79.1....}{.4.5....7}{1..6.....}%
{9368..1..}{..4...9..}{..7..4632}%
{.....8..1}{5....6.9.}{....4.82.}%
[DT2 tough 110]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TT110 (tough) \\
\sudoku{tt110.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tt115.sud}
\writepuzzle%
{432.5....}{.....3...}{75.1.....}%
{..5.823..}{.6.5.1.4.}{..193.2..}%
{.....6.58}{...3.....}{....2.934}%
[DT2 tough 115]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TT115 (tough) \\
\sudoku{tt115.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tt120.sud}
\writepuzzle%
{..8.1.7..}{.4.67.8..}{..2..461.}%
{.9....1.8}{.........}{2.1....9.}%
{.759..4..}{..4.52.6.}{..9.3.5..}%
[DT2 tough 120]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
TT120 (tough) \\
\sudoku{tt120.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st1130.sud}
\writepuzzle%
{8..356...}{5....4.1.}{34.......}%
{7.156..2.}{.6.....7.}{.2..796.5}%
{.......32}{.9.4....8}{...682..9}%
[Seattle Times 2005/11/30 3 stars]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
ST1130 (3 stars) \\
```

```
\sudoku{st1130.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st1207.sud}
\writepuzzle%
{...483..6}{.....57.2}{......38.}%
{..8..16..}{.26.5.17.}{..57..2..}%
{.82......}{9.18.....}{4..527...}%
[Seattle Times 2005/12/07  3 stars]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
ST1207 (3 stars) \\
\sudoku{st1207.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st1209.sud}
\writepuzzle%
{...6.....}{.4..8..5.}{.519..2.8}%
{..729...3}{.9.....1.}{4...679..}%
{6.8..957.}{.7..2..9.}{.....4...}%
[Seattle Times 2005/12/09  5 stars]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
ST1209 (5 stars) \\
\sudoku{st1209.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st1221.sud}
\writepuzzle%
{.....392.}{4....5.7.}{29.4...63}%
{..4....5.}{...2.1...}{.5....3..}%
{74...2.96}{.6.5....7}{.217.....}%
[Seattle Times 2005/12/21 3 stars]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
ST1221 (3 stars) \\
\sudoku{st1221.sud}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st1228.sud}
\writepuzzle%
{6..1..5..}{......379}{.942.....}%
{..5...8.3}{.4.....9.}{9.7...6..}%
{.....623.}{138......}{..6..8..7}%
[Seattle Times 2005/12/28 3 stars]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
ST1228 (3 stars) \\
\sudoku{st1228.sud}
\end{center}\end{minipage}
```

```
2474
2475 \renewcommand*{\puzzlefile}{st0126.sud}
2476 \writepuzzle%
2477 {..36...7.}{1...352..}{..4.....}%
2478 {..6....29}{7.......4}{32....5..}%
2479 {......1..}{..824...5}{.4...93..}%
2480 [Seattle Times 2006/01/26  4 stars]
2481 \vfill
2482 \begin{minipage}{0.95\linewidth}\begin{center}
2483 ST0126 (4 stars) \\
2484 \sudoku{st0126.sud}
2485 \end{center}\end{minipage}
2486
2487 \renewcommand*{\puzzlefile}{st0127.sud}
2488 \writepuzzle%
2489 {25.8.....}{....9.8.2}{9..4.7...}%
2490 {64.....9.}{.9.1.3.4.}{.7.....21}%
2491 {...5.4..7}{7.6.8....}{.....1.83}%
2492 [Seattle Times 2006/01/27  5 stars]
2493 \vfill
2494 \begin{minipage}{0.95\linewidth}\begin{center}
2495 ST0127 (5 stars) \\
2496 \sudoku{st0127.sud}
2497 \end{center}\end{minipage}
2498
2499 \renewcommand*{\puzzlefile}{st0204.sud}
2500 \writepuzzle%
2501 {.5..2.8.6}{.....6..3}{.6..132..}%
2502 {.7....96.}{4.......1}{.16....3.}%
2503 {..723..8.}{6..7.....}{2.4.9..7.}%
2504 [Seattle Times 2006/02/04 6 stars]
2505 \vfill
2506 \begin{minipage}{0.95\linewidth}\begin{center}
2507 ST0204 (6 stars) \\
2508 \sudoku{st0204.sud}
2509 \end{center}\end{minipage}
2510
2511 \renewcommand*{\puzzlefile}{st0208.sud}
2512 \writepuzzle%
2513 {21.9....7}{5.9..421.}{..7......}%
2514 {..875....}{3.14.95.8}{....361..}%
2515 {......4..}{.325..9.1}{1....8.25}%
2516 [Seattle Times 2005/02/08 3 stars]
2517 \vfill
2518 \begin{minipage}{0.95\linewidth}\begin{center}
2519 ST0208 (3 stars) \\
2520 \sudoku{st0208.sud}
2521 \end{center}\end{minipage}
2522
2523 \renewcommand*{\puzzlefile}{st0209.sud}
```

```
\writepuzzle%
{..9......}{.523.1...}{8.12.6...}%
{..4....1.}{97..2..84}{.2....9..}%
{...8.41.5}{...6.524.}{......8..}%
[Seattle Times 2005/02/09 4 stars]
\vfill
\begin{minipage}{0.95\linewidth}\begin{center}
ST0209 (4 stars) \\
\sudoku{st0209.sud}
\end{center}\end{minipage}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% answers

\clearpage
\pagestyle{empty}
\cleardoublepage
\pagestyle{myheadings}
\markboth{SOLUTIONS}{SOLUTIONS}

\cluefont{\normalsize}
\cellsize{1.5\baselineskip}

\renewcommand*{\puzzlefile}{se5.ans}
\writepuzzle%
{724195863}{561283497}{938647251}%
{189756324}{256314789}{473928615}%
{645839172}{892471536}{317562948}%
[SSBL easy 5 solution]
\renewcommand*{\puzzlefile}{se10.ans}
\writepuzzle%
{156349782}{482715639}{793286145}%
{649853271}{217964358}{835172496}%
{524697813}{371528964}{968431527}%
[SSBL easy 10 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
SE5 (easy solution) \\
\sudoku{se5.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
SE10 (easy solution) \\
\sudoku{se10.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{se15.ans}
\writepuzzle%
{456713982}{231698475}{879254631}%
{597846213}{124935867}{368172549}%
{643527198}{712489356}{985361724}%
```

```
[SSBL easy 15 solution]
\renewcommand*{\puzzlefile}{se20.ans}
\writepuzzle%
{368912547}{975438621}{412657839}%
{291543768}{783296414}{546781293}%
{657129384}{134865972}{829374156}%
[SSBL easy 20 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
SE15 (easy solution) \\
\sudoku{se15.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
SE20 (easy solution) \\
\sudoku{se20.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{se25.ans}
\writepuzzle%
{278963154}{645712839}{931854276}%
{823641795}{159387462}{764295381}%
{482579613}{397126548}{516438927}%
[SSBL easy 25 solution]
\renewcommand*{\puzzlefile}{se30.ans}
\writepuzzle%
{436129875}{128753496}{759864321}%
{514287639}{673591284}{892346517}%
{941638752}{365972148}{287415963}%
[SSBL easy 30 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
SE25 (easy solution) \\
\sudoku{se25.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
SE30 (easy solution) \\
\sudoku{se30.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tg5.ans}
\writepuzzle%
{764159382}{519238746}{823674951}%
{457982613}{936715428}{281463597}%
{142397865}{675841239}{398526174}%
[DT2 gentle 5 solution]
\renewcommand*{\puzzlefile}{tg10.ans}
\writepuzzle%
{382196574}{491573862}{567248139}%
```

```
{278614395}{946385217}{135729486}%
{624931758}{753862941}{819457623}%
[DT2 gentle 10 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
TG5 (gentle solution) \\
\sudoku{tg5.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
TG10 (gentle solution) \\
\sudoku{tg10.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tg15.ans}
\writepuzzle%
{462198573}{357462189}{981537426}%
{236741958}{175986342}{894253617}%
{723615894}{619824735}{548379261}%
[DT2 gentle 15 solution]
\renewcommand*{\puzzlefile}{tg20.ans}
\writepuzzle%
{256789431}{481326759}{739415628}%
{594261873}{167938245}{823574196}%
{912643587}{645897312}{378152964}%
[DT2 gentle 20 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
TG15 (gentle solution) \\
\sudoku{tg15.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
TG20 (gentle solution) \\
\sudoku{tg20.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tg25.ans}
\writepuzzle%
{782463519}{461295378}{395781246}%
{819527463}{634918752}{527346891}%
{256872934}{978134625}{243659187}%
[DT2 gentle 25 solution]
\renewcommand*{\puzzlefile}{tg30.ans}
\writepuzzle%
{376418529}{928756134}{154923678}%
{749132865}{612584793}{835697241}%
{597361482}{483275916}{261849357}%
[DT2 gentle 30 solution]
\vfill
```

```
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
TG25 (gentle solution) \\
\sudoku{tg25.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
TG30 (gentle solution) \\
\sudoku{tg30.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st1226.ans}
\writepuzzle%
{962854731}{745213689}{183967452}%
{326748915}{571639248}{498125376}%
{837592164}{659471823}{214386597}%
[Seattle Times 2005/12/26 1 star solution]
\renewcommand*{\puzzlefile}{st0206.ans}
\writepuzzle%
{176429385}{352817496}{849653217}%
{463971852}{925384671}{718265934}%
{281536749}{534792168}{697148523}%
[Seattle Times 2005/02/06 1 star solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
ST1226 (1 star solution) \\
\sudoku{st1226.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
ST0206 (1 star solution) \\
\sudoku{st0206.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st0123.ans}
\writepuzzle%
{594621837}{736489125}{281357469}%
{319278654}{652134978}{478596312}%
{927813546}{145962783}{863745291}%
[Seattle Times 2006/01/23 1 star solution]
\renewcommand*{\puzzlefile}{cs1.ans}
\writepuzzle%
{546987312}{213654978}{879321645}%
{657198423}{324765189}{981432756}%
{435876291}{768219534}{192543867}%
[CreateSudoku 1 (level 2) solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
ST0123 (1 star solution) \\
\sudoku{st0123.ans}
\end{center}\end{minipage}
```

```
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
CS1 (level 2 solution) \\
\sudoku{cs1.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{cs2.ans}
\writepuzzle%
{978312654}{312645987}{645978321}%
{189423765}{423756198}{756189432}%
{291534876}{867291543}{534867219}%
[CreateSudoku 2 (level 2) solution]
\renewcommand*{\puzzlefile}{cs3.ans}
\writepuzzle%
{456897123}{123564789}{789231456}%
{234675891}{567918234}{891342567}%
{345786912}{912453678}{678129345}%
[CreateSudoku 3 (level 2) solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
CS2 (level 2 solution) \\
\sudoku{cs2.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
CS3 (level 2 solution) \\
\sudoku{cs3.ans}
\end{center}\end{minipage}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% medium

\renewcommand*{\puzzlefile}{sm5.ans}
\writepuzzle%
{387691245}{546273198}{921485637}%
{865342971}{193567482}{274918356}%
{618739524}{759124863}{432856719}%
[SSBL medium 5 solution]
\renewcommand*{\puzzlefile}{sm10.ans}
\writepuzzle%
{635748219}{418392576}{279561483}%
{527984631}{941653827}{863127954}%
{194835762}{786219345}{352476198}%
[SSBL medium 10 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
SM5 (medium solution) \\
\sudoku{sm5.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
```

```
SM10 (medium solution) \\
\sudoku{sm10.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sm15.ans}
\writepuzzle%
{853974612}{471326958}{962815473}%
{396542781}{745681239}{218739564}%
{134267895}{587493126}{629158347}%
[SSBL medium 15 solution]
\renewcommand*{\puzzlefile}{sm20.ans}
\writepuzzle%
{863452197}{594178326}{172639548}%
{617894235}{938526471}{234713869}%
{489265713}{351947682}{726381954}%
[SSBL medium 20 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
SM15 (medium solution) \\
\sudoku{sm15.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
SM20 (medium solution) \\
\sudoku{sm20.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sm25.ans}
\writepuzzle%
{382467951}{716935284}{495218376}%
{564329817}{278651493}{139784625}%
{927146538}{651893742}{843572169}%
[SSBL medium 25 solution]
\renewcommand*{\puzzlefile}{sm30.ans}
\writepuzzle%
{765834921}{318279465}{294615378}%
{657392814}{823741659}{941568732}%
{439157286}{186423597}{572986143}%
[SSBL medium 30 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
SM25 (medium solution) \\
\sudoku{sm25.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
SM30 (medium solution) \\
\sudoku{sm30.ans}
\end{center}\end{minipage}

```

```
\renewcommand*{\puzzlefile}{tm35.ans}
\writepuzzle%
{591328746}{674951823}{238674195}%
{356297481}{427185369}{819463257}%
{985732614}{742516938}{163849572}%
[DT2 moderate 35 solution]
\renewcommand*{\puzzlefile}{tm40.ans}
\writepuzzle%
{874136592}{952478136}{316592874}%
{631284759}{287915463}{495367281}%
{763841925}{548629317}{129753648}%
[DT2 moderate 40 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
TM35 (moderate solution) \\
\sudoku{tm35.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
TM40 (moderate solution) \\
\sudoku{tm40.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tm45.ans}
\writepuzzle%
{716248935}{538769421}{924531786}%
{179683542}{652974813}{483152679}%
{241896357}{897325164}{365417298}%
[DT2 moderate 45 solution]
\renewcommand*{\puzzlefile}{tm50.ans}
\writepuzzle%
{751962834}{468375219}{392148756}%
{627891345}{143526987}{589437162}%
{975613428}{834259671}{216784593}%
[DT2 moderate 50 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
TM45 (moderate solution) \\
\sudoku{tm45.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
TM50 (moderate solution) \\
\sudoku{tm50.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tm55.ans}
\writepuzzle%
{731946528}{245138679}{968572143}%
{614729385}{572813964}{893465712}%
```

```
{159684237}{426357891}{387291456}%
[DT2 moderate 55 solution]
\renewcommand*{\puzzlefile}{tm60.ans}
\writepuzzle%
{734198562}{156324798}{982567431}%
{813752649}{465983217}{279416853}%
{341875926}{697241385}{528639174}%
[DT2 moderate 60 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
TM55 (moderate solution) \\
\sudoku{tm55.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
TM60 (moderate solution) \\
\sudoku{tm60.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st0914.ans}
\writepuzzle%
{567293184}{298461537}{413578692}%
{321759468}{649812753}{785634921}%
{936127845}{872945316}{154386279}%
[Seattle Times 2005/09/14 2 star solution]
\renewcommand*{\puzzlefile}{st0207.ans}
\writepuzzle%
{872351946}{561849273}{934672581}%
{759283164}{628514397}{413967852}%
{285436719}{196725438}{347198625}%
[Seattle Times 2005/02/07 2 star solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
ST0914 (2 star solution) \\
\sudoku{st0914.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
ST0207 (2 star solution) \\
\sudoku{st0207.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{cs4.ans}
\writepuzzle%
{123465978}{456798312}{789132645}%
{891243756}{234576189}{567819423}%
{912354867}{678921534}{345687291}%
[CreateSudoku 4 (level 4) solution]
\renewcommand*{\puzzlefile}{cs5.ans}
\writepuzzle%
```

```
{231654978}{564987312}{897321645}%
{918432756}{675198423}{342765189}%
{786219534}{453876291}{129543867}%
[CreateSudoku 5 (level 4) solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
CS4 (level 8 solution) \\
\sudoku{cs4.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
CS5 (level 8 solution) \\
\sudoku{cs5.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{cs6.ans}
\writepuzzle%
{312654978}{645987312}{978321645}%
{756198423}{423765189}{189432756}%
{534876291}{291543867}{867219534}%
[CreateSudoku 6 (level 4) solution]
\renewcommand*{\puzzlefile}{sh5.ans}
\writepuzzle%
{269175438}{471832965}{583694217}%
{715426389}{894317526}{326958741}%
{942561873}{138749652}{657283194}%
[SSBL hard 5 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
CS6 (level 4 solution) \\
\sudoku{cs6.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
SH5 (hard solution) \\
\sudoku{sh5.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sh10.ans}
\writepuzzle%
{437921865}{156837492}{289456137}%
{598163274}{374295681}{621748953}%
{713682549}{845319726}{962574318}%
[SSBL hard 10 solution]
\renewcommand*{\puzzlefile}{sh15.ans}
\writepuzzle%
{216935847}{478261359}{395478261}%
{761392584}{952846713}{843157926}%
{124783695}{539624178}{687519432}%
[SSBL hard 15 solution]
```

```
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
SH10 (hard solution) \\
\sudoku{sh10.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
SH15 (hard solution) \\
\sudoku{sh15.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sh20.ans}
\writepuzzle%
{468271935}{925438671}{713695284}%
{154762893}{832519467}{679384512}%
{587943126}{296157348}{341826759}%
[SSBL hard 20 solution]
\renewcommand*{\puzzlefile}{sh25.ans}
\writepuzzle%
{439561728}{167238459}{852497613}%
{286953174}{794816532}{513724986}%
{348179265}{621345897}{975682341}%
[SSBL hard 25 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
SH20 (hard solution) \\
\sudoku{sh20.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
SH25 (hard solution) \\
\sudoku{sh25.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{sh30.ans}
\writepuzzle%
{461258793}{253917846}{789463152}%
{346789521}{572146389}{918325674}%
{894671235}{635892417}{127534968}%
[SSBL hard 30 solution]
\renewcommand*{\puzzlefile}{tt100.ans}
\writepuzzle%
{513896427}{742315698}{968742531}%
{154289376}{376154289}{289637154}%
{891563742}{425971863}{637428915}%
[DT2 tough 100 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
SH30 (hard solution) \\
\sudoku{sh30.ans}
```

```
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
TT100 (tough solution) \\
\sudoku{tt100.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tt105.ans}
\writepuzzle%
{925378146}{681254793}{734916582}%
{256739418}{178462359}{349581267}%
{597843621}{462197835}{813625974}%
[DT2 tough 105 solution]
\renewcommand*{\puzzlefile}{tt110.ans}
\writepuzzle%
{379412586}{648539217}{125687349}%
{936825174}{214763958}{857194632}%
{492358761}{581276493}{763941825}%
[DT2 tough 110 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
TT105 (tough solution) \\
\sudoku{tt105.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
TT110 (tough solution) \\
\sudoku{tt110.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{tt115.ans}
\writepuzzle%
{432758196}{196243587}{758169423}%
{945682371}{263571849}{871934265}%
{329416758}{584397612}{617825934}%
[DT2 tough 115 solution]
\renewcommand*{\puzzlefile}{tt120.ans}
\writepuzzle%
{968215734}{143679852}{752384619}%
{496523178}{537198246}{281746395}%
{875961423}{314852967}{629437581}%
[DT2 tough 120 solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
TT115 (tough solution) \\
\sudoku{tt115.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
TT120 (tough solution) \\
```

```
\sudoku{tt120.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st1130.ans}
\writepuzzle%
{819356247}{576924813}{342817596}%
{781563924}{965248371}{423179685}%
{658791432}{297435168}{134682759}
[Seattle Times 2005/11/30 3 star solution]
\renewcommand*{\puzzlefile}{st1207.ans}
\writepuzzle%
{217483956}{834695712}{659172384}%
{798231645}{326954178}{145768239}%
{582319467}{971846523}{463527891}%
[Seattle Times 2005/12/07 3 star solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
ST1130 (3 star solution) \\
\sudoku{st1130.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
ST1207 (3 star solution) \\
\sudoku{st1207.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st1209.ans}
\writepuzzle%
{839652147}{246781359}{751943268}%
{167295483}{592438716}{483167925}%
{628319574}{374526891}{915874632}%
[Seattle Times 2005/12/09 5 star solution]
\renewcommand*{\puzzlefile}{st1221.ans}
\writepuzzle%
{517863924}{436925178}{298417563}%
{174389652}{683251749}{952674381}%
{745132896}{369548217}{821796435}%
[Seattle Times 2005/12/21 3 star solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
ST1209 (5 star solution) \\
\sudoku{st1209.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
ST1221 (3 star solution) \\
\sudoku{st1221.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st1228.ans}
```

```
\writepuzzle%
{673189524}{812654379}{594273186}%
{265791843}{341865792}{987342651}%
{795416238}{138927465}{426538917}%
[Seattle Times 2005/12/28 3 star solution]
\renewcommand*{\puzzlefile}{st0126.ans}
\writepuzzle%
{253698471}{197435286}{864127953}%
{416573829}{785912634}{329864517}%
{972356148}{638241795}{541789362}%
[Seattle Times 2006/01/26 4 star solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
ST1228 (3 star solution) \\
\sudoku{st1228.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
ST0126 (4 star solution) \\
\sudoku{st0126.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st0127.ans}
\writepuzzle%
{253816479}{467395812}{918427536}%
{641752398}{892163745}{375948621}%
{189534267}{736289154}{524671983}%
[Seattle Times 2006/01/27 5 star solution]
\renewcommand*{\puzzlefile}{st0204.ans}
\writepuzzle%
{153427896}{742986513}{869513247}%
{375142968}{428369751}{916875432}%
{597321684}{681754329}{234698175}%
[Seattle Times 2006/02/04 6 star solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
ST0127 (5 star solution) \\
\sudoku{st0127.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
ST0204 (6 star solution) \\
\sudoku{st0204.ans}
\end{center}\end{minipage}

\renewcommand*{\puzzlefile}{st0208.ans}
\writepuzzle%
{213985647}{589674213}{647213859}%
{928751364}{361429578}{754836192}%
{875192436}{432567981}{196348725}%
```

```
[Seattle Times 2005/02/08 3 star solution]
\renewcommand*{\puzzlefile}{st0209.ans}
\writepuzzle%
{639587421}{752341698}{841296753}%
{584769312}{976123584}{123458976}%
{297834165}{318675249}{465912837}%
[Seattle Times 2005/02/09 4 star solution]
\vfill
\noindent\begin{minipage}{0.47\linewidth}\begin{center}
ST0208 (3 star solution) \\
\sudoku{st0208.ans}
\end{center}\end{minipage}
\hfill
\begin{minipage}{0.47\linewidth}\begin{center}
ST0209 (4 star solution) \\
\sudoku{st0209.ans}
\end{center}\end{minipage}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% bibliography
\cleardoublepage

\bibliographystyle{alpha}
\begingroup
\raggedright
\begin{thebibliography}{GMSN94A}

\bibitem[Mep05]{DT2}
Michael Mepham.
\emph{The Daily Telegraph Sudoku 2}.
\newblock Pan Books, 2005.
\newblock ISBN 0-330-44203-3.

\bibitem[Wil06]{SUDOKUBUNDLE}
Peter Wilson.
\emph{The sudoku bundle for displaying, solving and generating
      Sudoku puzzles}.
\newblock February 2006.
\newblock (Available from CTAN in \url{macros/latex/contrib/sudokubundle}).

\bibitem[SOL]{SOL}
\emph{Sudoku Online: Home of the Sudokulist}.
\newblock \url{http://www.sudoku.org.uk}

\bibitem[SSBL]{SSBL}
\emph{Sudoku Solver ... by logic}.
\newblock \url{http://www.sudokusolver.co.uk}

\end{thebibliography}
\endgroup
```

```
3224
3225 \end{document}
3226
3227 ⟨/examples⟩
```

References

[Abr05] Paul Abraham. *The sudoku package*. June 2005. (Available from CTAN in `macros/latex/contrib/sudoku`)

[Ars95] Donald Arseneau. *Generating random numbers in TEX*. 1995. (Available from CTAN in `macros/generic/random.tex`)

[MG04] Frank Mittelbach and Michel Goossens. *The LaTeX Companion*. Second edition. Addison-Wesley Publishing Company, 2004. ISBN 0-201-36299-6.

[PJ05] TEX Users Group. Distractions: Sudoku. *The PracTeX Journal*. 2005-4. `http://tug.org/pracjourn2005-4/distract`

[SOL] *Sudoku Online: Home of the Sudokulist*. `http://www.sudoku.org.uk`

[SSBL] *Sudoku Solver ... by logic*. `http://www.sudokusolver.co.uk`

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

S

T

U