

Standard Puzzle Format

Specification

1. Abstract

This specification defines the Standard Puzzle Format (SPF). SPF is a format to describe the diagrams used for logical puzzles (such as sudoku, fences and tapa). The format is designed to be human readable as well as accessible by computers. It is embedded in the postscript language, which makes it easy to display and print the puzzle diagrams.

2. General description

The general layout of the format is given as follows:

```
prolog
<<
key-value-pair*
>>
epilog
```

A file in SPF-format consists of a prolog, an EPS dictionary and an epilog. The prolog and the epilog can be anything, with the only exception, that the prolog might not contain the sequence <<. The dictionary starts with << and ends with >>. The >> must be preceded by a newline character. Between this two delimiters, the dictionary consists of key-value-pairs.

The key always starts at the beginning of a line, that is it is preceded by a newline, and its first character is a slash (/), immediately followed by the name of the key. Key and value are separated by a space. The key must not contain a space and should consist only of letters (lower and uppercase, which are considered different), digits and underscores.

3. Some special keys

The dictionary has to contain a key called `/type` with the value being a postscript string containing the name of the puzzletype. This type defines the puzzle specific entries of the dictionary. If there are several formats, the `/format` can be specified as a version number. If no format is given, it can be assumed to be 1.

The key `/sol` should also be defined as a boolean value (true or false). This defines, whether the puzzle or the solution of the puzzle should be shown. Of

course, this only matters, if the puzzle is to be displayed.

The key `/author` is used to specify the author of the puzzle as a postscript string. A description of the puzzle can be given using the `/description` key. The value is an array of strings.

In most cases, the puzzle diagram is given as an ASCII art image as value of the key `/puzzle` and the solution is given as value of the key `/solution`. In both cases, the type of the values is almost always an array of strings.

As some applications may need to know the size of the diagram, before they read it, the size should also be given, either as `X` and `Y` or as `size` if the diagram has to be a square.

4. An example

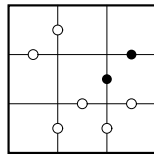
The example given on the next page defines the kropki puzzle given on the next page.

The prolog consists of some postscript comments which are typical of encapsulated postscript files (EPS). The three dots indicate, that more postscript code can be added at this place. We'll soon have a closer look at the dictionary that follows. After the dictionary more postscript code can follow.

The dictionary defines five key-value-pairs with keys `/type`, `/sol`, `/size`, `/puzzle` and `/description`. The value of the type-key is "(kropki)". The parenthesis are used in postscript to describe a string literal. The value of the sol-key is false, as the file should not show the solution. Indeed no solution is given, which is fine, if the value of the sol-key is false.

Next, the size of the diagram is specified. As kropkis are always squares, the size is given with the size-key and has the value of 3. The value associated with the puzzle-key is the main puzzle. It is defined as an array – the brackets, which contains some strings. The strings are arranged to make up for a nice ASCII art of the puzzle.

Last, a description follows, which is in this case only one line: "size 3 times 3". The description should be stuff, the user might know, before he starts to solve the puzzle. In this case the size is given in human readable form.



```

%!PS-Adobe-3.0 EPSF-3.0
%%Pages: 1
%%BoundingBox: 0 0 70 70
%%CropBox: 0 0 70 70
%%DocumentData: Clean7Bit
%%Page: 1 1
%%EndComments
...
<<
/type (kropki)
/sol false
/size 3
/puzzle [
(+--+--+ )
(| 0 | )
(+0+ +++)
(| * | )
(+ +0+0+)
(| 0 0 | )
(+--+--+ ) ]
/description [
(size 3 times 3) ]
>> currentdict copy pop
...
%%EOF

```

Appendix

A. Specification of some puzzle types

In the following puzzle specific specification the words vertice, edge and cell are used to describe the intersection of the lines, the line segment between such intersections and the area inside of line segments.

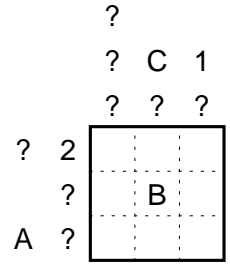
A.1. ABC-Box

```

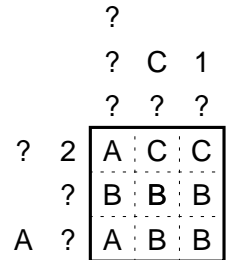
/X 3
/Y 3
/max 3
/left 2
/top 3
/digits 1
/puzzle [
( ? )
( ?C1)
( ???)
( +---)
(?2| )
( ?| B )
(A?| ) ]
/solution [
(ACC)
(BBB)
(ABB) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional max-key can be given, which denotes the number of letters, starting at the beginning of the alphabet, that can appear in the puzzle. If omitted, 3 is assumed. The left-key and the top-key give the maximum number of numbers at the left and at the top of the puzzle. They should be given, as some programmes might need them to know in advance the size of the puzzle-key. An optional digits-key can be given, which denotes the maximum number of digits of numbers outside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $x + dl + 1$ and the height of the diagram is $y + dt + 1$, where x is the width of the puzzle, y is the height of the puzzle, l and t are the values of the left- and top-key and d is the value of the digits-key.

The place $dl \times dt$ is marked with a plus. Right of that plus are minuses and below that plus are pipes. They primarily serve to help humans, but the plus might be used by programmes to decide upon the values of the left- and top-key if they are omitted. Minuses and pipes might be omitted.

Inside the diagram, every cell of the puzzle contains a space or a letter, which has to be in the range given by the max-key.

Outside the diagram, the hints are given. The content can be a question mark, a letter in the range specified by the max-key or a number greater or equal to 1. Rows or columns might also be completely

empty, which denotes, that there is no given information about this row or column. If the digits-key is greater than 1, the numbers at the top are written from top down.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the cells is given as single character.

A.2. ABCD

```

/X 3
/Y 3
/max 4
/digits 1
/puzzle [
(      )
(    2 0)
(      0 )
(      12)
(    +---)
(    |   )
(0   |B #)
(    |   ) ]
/solution [
(CDA)
(BA#)
(CDA) ]

```

Puzzle:

	2			0	
			0		
				1	2
0				B	

Solution:

	2			0	
			0		
				1	2
0				B	A
				C	D

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional max-key can be given, which denotes the maximum letter in the alphabet, that can appear in the puzzle. If omitted, 4 is assumed. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $x + dm + 1$ and the height of the diagram is $y + dm + 1$, where x is the width of the puzzle, y is the height of the puzzle, d is the value of the digits-key and m is the value of the max-key.

The place $2dm \times 2dm$ is marked with a plus. Right of that plus are minuses and below that plus are pipes. They primarily serve to help humans, but the plus might be used by programmes to decide upon the value of the max-key if omitted. Minuses and pipes might be omitted.

Inside the diagram, every cell of the puzzle contains one character, which can be a letter, a space or a hash key, the later one denoting a field that is blackend.

Outside the diagram, the hints are given. At the left they are as many charakters wide, as the digits-key gives. At the top they are as many charakters high, as the digits-key gives. Numbers have to be right or bottom aligned.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the inner cells is given as single character.

A.3. ABCD Diagonal

Same as ABCD, but the default for the max key is 5.

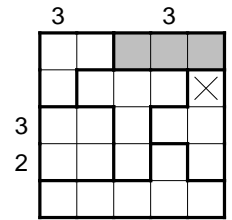
A.4. Aquarium

```

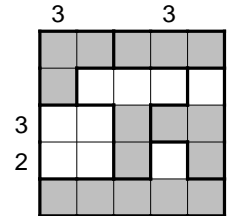
/X 5
/Y 5
/digits 2
/puzzle [
(
( 3 3 )
( +-+--+--+ )
( | |# # #| )
( + +-+--+--+ )
( | | | -| )
( +-+--+ +-+ )
( 3| | | | )
( + + + +-+ )
( 2| | | | )
( +-+--++ +-+ )
( | | | )
( +-+--+--+ ) ]
/solution [
(#####)
( # )
( ###)
( # #)
(#####) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers outside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x + d + 1$ and the height of the diagram is $2y + d + 1$, where x and y are the width and the height of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

Inside the diagram, the edges between two areas are marked by pipes and minuses as well. In the cells, # and - can be given to denote a filled or an empty cell.

At the left and the top numbers can be given. Numbers at the left have to be right alligned. Numbers at the top are written top-down and have to be bottom alligned.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the cells is given.

A.5. Arukone

```

/X 5
/Y 5
/puzzle [
(+--+--+--+--++)
(|      C  |)
(+ + + + +)
(|      #|)
(+ + + + +)
(|  B A  |)
(+ + + + +)
(| A    B|)
(+ + + + +)
(|  C  - |)
(+--+--+--+--++) ]
/solution [
(+--+--+--+--++)
(| - - -C |)
(+|+ + + +)
(| - -   |)
(+|+|+ +|+ +)
(|  B A  |)
(+|+|+|+ + +)
(| A  - B|)
(+|+ + +|+|+)
(| - -C  - |)
(+--+--+--+--++) ]

```

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given. If given, numbers are used instead of letters. The key gives the maximum number of digits of these numbers.

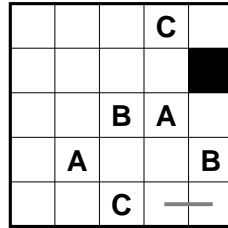
The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2dx + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

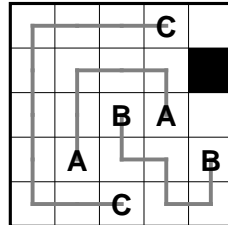
The cells may contain an uppercase letter or # which is a cell, that must not be used by a path. In case of numbers, the cell may contain a number, which has to be right aligned. Between two cells, there might be minus and pipe symbols to show part of the path. The pipe symbol has to be centered, if not possible it has to be slightly right of the center.

The solution-key is the same as the puzzle-key.

Puzzle:



Solution:



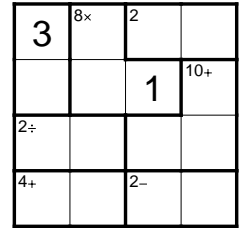
A.6. Basic

```

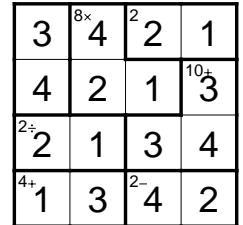
/size 4
/digits 2
/puzzle [
(+--+--+--+--++)
(|3 | x|   |)
(| | 8| 2  |)
(+ + +--+--++)
(| |  1 | +|)
(| |    |10|)
(+--+--+--+ +)
(| :  |   |)
(| 2  |   |)
(+--+--+--+--++)
(| +  | -  |)
(| 4  | 2  |)
(+--+--+--+--++) ]
/solution [
(3421)
(4213)
(2134)
(1342) ]

```

Puzzle:



Solution:



The size-key defines the size of the puzzle. It has to be greater or equal to 1 and less or equal to 9. An optional digits-key can be given, which denotes the maximum number of digits of small numbers inside the diagram. If omitted or less than 2, it is assumed to be 2.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $(d + 1)s + 1$ and the height of the diagram is $3s + 1$, where s is the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

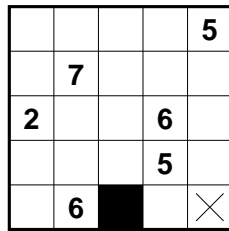
Inside the diagram, the edges between two areas are marked by pipes and minuses as well. In the cells, at the top left a number can be given, which is a large number in this cell. At the top right an operator can be given, which is +, -, x, : or a space. In the bottom row a right aligned number can be given.

The solution-key gives the solution as ascii art. In the solution diagram, only the numbers are given.

A.7. Cave

```
/X 5
/Y 5
/digits 2
/puzzle [
(          5)
(   7     )
(  2    6  )
(          5)
(   6##  --) ]
/solution [
(          )
( #   # )
(  #   )
( # # # )
(  #   ) ]
```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is dx and the height of the diagram is y , where x and y are the size of the puzzle and d is the value of the digits-key.

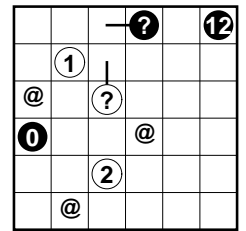
Cells can contain # to denote part of the walls, - to denote part of the cave or a number, which has to be right aligned.

The solution-key has one symbol per cell. The hash-symbol for walls and a space else.

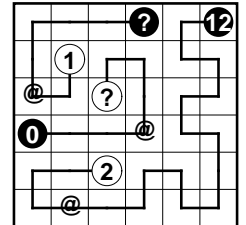
A.8. Chatroom

```
/X 6
/Y 6
/digits 2
/puzzle [
(+-----+-----+-----+-----+)
(|           - ?#      12#|)
(+  +  +  +  +  +  +  +)
(|    1                |)
(+  +  + | +  +  +  +)
(| @     ?              |)
(+  +  +  +  +  +  +)
(| 0#           @        |)
(+  +  +  +  +  +  +)
(|           2            |)
(+  +  +  +  +  +  +)
(|    @                   |)
(+-----+-----+-----+-----+) ]
/solution [
(+--+--+--+--+--+--+)
(| - - -*  -*|)
(+|+ + + +|+ +)
(| * - - |)
(+|+|+|+|+ +|+)
(|@- * - |)
(+ + + +|+|+ +)
(|*- - -@ - |)
(+ + + + +|+)
(| - -* - |)
(+|+ + +|+|+|+)
(| -@- - - |)
(+--+--+--+--+--+--+) ]
```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $(d+2)x+1$ and the height of the diagram is $2y+1$, where x and y are the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

The cells are $d+1$ characters wide. If the rightmost character is the hash-key, the symbol in that cell is black. Else it's white. The rest of the cell contains either the symbol |@| which is a phone, or a question-mark or a number, which has to be right aligned.

Between two cells, there might be minus and pipe symbols to show part of the path.

The solution-key gives the puzzle as ascii art. The width of the diagram is $2x + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

The cells are one character wide. And might be completely empty. For better reading, the phones might be shown as @ and the numbers as *.

Between two cells, there are minus and pipe symbols to show the solution.

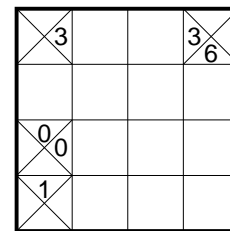
A.9. Compass

```

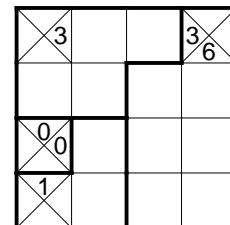
/X 4
/Y 4
/digits 2
/puzzle [
(.. 3      ....)
(....      3 6)
(          )
(          )
( 0 0      )
(....      )
( 1..      )
(....      ) ]
/solution [
(+--+--+ +)
(|      | |)
(+ + +--+ +)
(|      | |)
(+--+ + + +)
(| | | | |)
(+--+ + + +)
(|      | |)
(+--+--+--+ +) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2dx$ and the height of the diagram is $2y$, where x and y are the size of the puzzle and d is the value of the digits-key.

The cells of the puzzle are given as blocks of size $2d \times 2$, that contain the clues. The first clue is the top value, the second clue is the right value, the third clue is the left value and the last clue is the bottom value. Clues can be numbers, right-aligned, or dots for empty clues.

The solution-key gives the puzzle as ascii art. The width of the diagram is $2x + 1$ and the height of the diagram is $2y + 1$.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

Inside the diagram, the edges between two areas are marked by pipes and minuses as well.

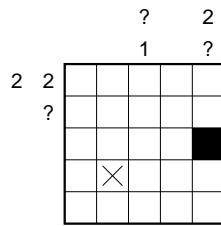
A.10. Coral

```

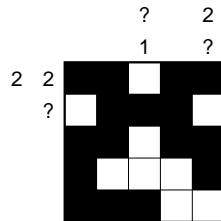
/X 5
/Y 5
/left 2
/top 2
/digits 2
/puzzle [
(      )
(      ? 2)
(      )
(      1 ?)
(  +-----)
( 2 2|   )
(  ?|   )
(  |   #)
(  | - )
(  |   ) ]
/solution [
(## ##)
( ### )
(## ##)
(#  #)
(### ) ]

```

Puzzle:



Solution:



to upper.

Rows or columns might also be completely empty, which denotes, that there is no given information about this row or column.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the cells is given.

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $x + dl + 1$ and the height of the diagram is $y + dt + 1$, where x is the width of the puzzle, y is the height of the puzzle, l and t are the values of the left- and top-key and d is the value of the digits-key.

The place $dl \times dt$ is marked with a plus. Right of that plus are minuses and below that plus are pipes. They primarily serve to help humans, but the plus might be used by programs to decide upon the values of the left- and top-key if they are omitted. Minuses and pipes might be omitted.

Inside the diagram, every cell of the puzzle contains one character, which can be a # to denote part of the coral and - to a cell which is not part of the coral.

Outside the diagram, the hints are given. At the left they are as many characters wide, as the digits-key gives. At the top they are as many characters high, as the digits-key gives. Numbers have to be right or bottom aligned. Instead of a number, a questionmark can be used. The order of the numbers can be arbitrary, but as a convenience they should be ordered: First the questionmarks, than the number vom lower

A.11. Curve Data

```

/X 4
/Y 4
/digits 2
/puzzle [
(+---+---+---+)
(| 1           2|)
(+  +  +  +  +)
(|           3  |)
(+ |+  +  +  +)
(|  ##  -  |)
(+  +  +  +  +)
(|           |)
(+---+---+---+) ]
/shapes [ [
( + )
(| )
(+--) ] [
(+--) ] [
( +-+)
(  |)
(+---+)
(|  |)
(+---+) ] ]
/solution [
(+--+--+--++)
(|+ +---+|)
(+|. . . +)
(|| +---+|)
(+|. . . |+)
(|| # +--+|)
(+|. .|. |+)
(|+-+ +--+|)
(+--+--+--++) ]

```

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

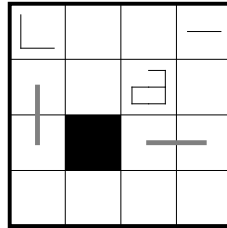
The puzzle-key gives the puzzle as ascii art. The width of the diagram is $(d + 1)x + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

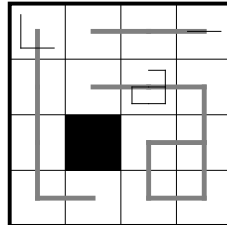
Inside the diagram, cells can contain a number, which has to be right aligned or hash symbols, where all places inside of the cell have to be filled with this symbol.

Between two cells, there might be minus and pipe

Puzzle:



Solution:



symbols to show part of the path.

The shapes-key is an array, where each entry corresponds to the number in the diagram. Array entries start with number 1. Each entry gives a shape in ascii art. Each shape consists of vertices marked with a plus symbol and edges marked with minus and pipe symbols. Vertices may only occur at places that are in odd rows and columns.

Edges must end in a vertex and the whole graph of one shape must be connected. Vertices must not touch orthogonally. There should not be any remaining white space at the borders of a shape, but programs must not rely on this. Shapes must not contain abnormal vertices, that is, vertices, where the shape neither dissects nor bends nor ends.

The solution-key gives the solution as ascii art. The width of the diagram is $2x + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle.

All vertices at the border are marked as plus symbols, all other vertices are marked as dots. The edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

Inside the diagram, cells can contain a hash symbol which denotes a black cell or a plus symbol which denotes a vertex of a curve. The edges of the curves are given by minus and pipe symbols between cells. If an edge of the graph goes straight through a cell, that cell must also contain a minus or a pipe symbol.

A.12. Domino Hunt

```

/X 5
/Y 5
/min 1
/max 4
/digits 2
/puzzle [
(+---+---+---+---+)
(|## 2 3 2 ##|)
(+ + + + + +)
(| 4 4 3 4 3|)
(+ + + + + +)
(| 2 ## 3 ## 4|)
(+ + + + + +)
(| 2 1 1 1 4|)
(+ + + + + +)
(| 1 2 1 3 ##|)
(+---+---+---+---+) ]
/solution [
(+---+---+---+---+)
(|##| 2 3| 2|##|)
(+---+---+---+---+)
(| 4 4| 3| 4| 3|)
(+---+---+---+---+)
(| 2|##| 3|##| 4|)
(+ +---+---+---+---+)
(| 2| 1 1| 1 4|)
(+---+---+---+---+)
(| 1 2| 1 3|##|)
(+---+---+---+---+) ]

```

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

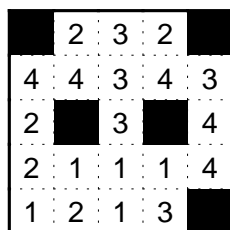
The puzzle-key gives the puzzle as ascii art. The width of the diagram is $(d+1)x+1$ and the height of the diagram is $2y+1$, where x and y are the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

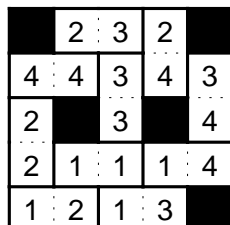
Inside the diagram, cells can contain a number, which has to be right aligned or hash symbols, where all places inside of the cell have to be filled with this symbol. Horizontal and vertical lines, that separate, can be given as minuses and pipes.

The solution-key is the same as the puzzle-key.

Puzzle:



Solution:



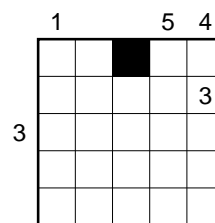
A.13. Doubleblock

```

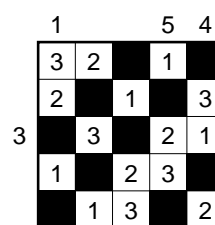
/size 5
/digits 2
/puzzle [
( 1 5 4)
( +-----)
( | ## )
( | 3)
( 3| )
( | )
( | ) ]
/solution [
( 3 2## 1##)
( 2## 1## 3)
(## 3## 2 1)
( 1## 2 3##)
(## 1 3## 2) ]

```

Puzzle:



Solution:



The size-key defines the size of the puzzle. It has to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $d(x+1)+1$ and the height of the diagram is $y+2$, where x and y are the size of the puzzle and d is the value of the digits-key.

Each cell can contain a number, which is right aligned, be completely filled with hash-symbols to denote a block or be blank. At the left and the top numbers can be given. They also have to be right aligned.

The solution-key is the same as the puzzle-key, but only the content of the cells is given.

A.14. Easy as ABC

```

/size 5
/max 3
/puzzle [
( C )
(A - )
( B )
( B B)
( )
(A )
( AA ) ]
/solution [
(ABC )
( C AB)
(CAB )
(B CA)
( ABC) ]

```

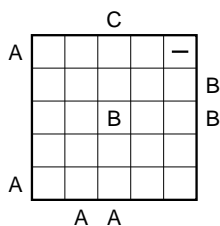
The size-key defines the size of the puzzle. It has to be greater or equal to 1. The max-key gives the position in the alphabet of the last used letter.

The puzzle-key gives the puzzle as ascii art. The width and height of the diagram are $s + 2$, where s is the size of the puzzle.

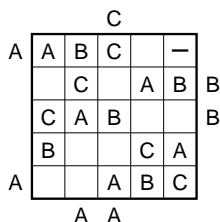
The letters at the border are given in the leftmost, rightmost column and top and bottom row. In the center, letters and minus-signs can be given. A minus sign denotes a cell, that cannot be occupied by a letter.

The solution-key gives the solution as ascii art. In the solution diagram, only the center is given.

Puzzle:



Solution:



A.15. Fences

There are two formats defined for the fences puzzle: Format 1 is a special format for rectangular grids, than might contain holes of size 2×2 , but not larger. Format 2 is a more general format, that can handle arbitrary planar graphs.

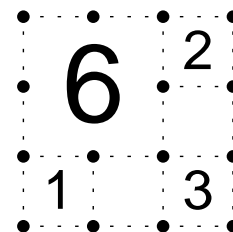
A.15.1. Format 1

```

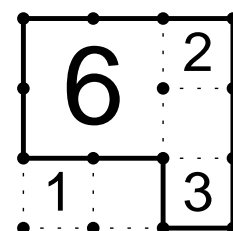
/X 3
/Y 3
/puzzle [
(+--+--+ )
(| |2| )
(+ 6 +--+ )
(| | | )
(+--+--+ )
(|1| |3| )
(+--+--+ ) ]
/solution [
(+--+--+ )
(| |2| )
(+ 6 + + )
(| | | )
(+--+ + )
( 1 |3| )
(+ + +--+ ) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle.

All vertices are marked as plus symbols, all edges, that belong to the grid (that is, all edges, that are not inside the hole) are marked by pipes and minuses.

In the center of all cells, a one digit number can be given.

The solution-key is the same as the puzzle-key, but only the edges, that are part of the solution are marked by minuses and pipes.

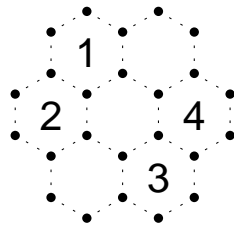
A.15.2. Format 2

```

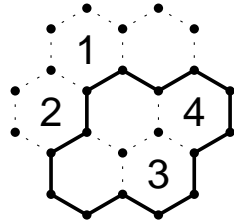
/node [
[ 0.693 0.4 ]
[ 1.386 0 ]
[ 2.078 0.4 ]
[ 2.771 0 ]
[ 3.464 0.4 ]
[ 0 1.6 ]
[ 0.693 1.2 ]
[ 1.386 1.6 ]
[ 2.078 1.2 ]
[ 2.771 1.6 ]
[ 3.464 1.2 ]
[ 4.157 1.6 ]
[ 0 2.4 ]
[ 0.693 2.8 ]
[ 1.386 2.4 ]
[ 2.078 2.8 ]
[ 2.771 2.4 ]
[ 3.464 2.8 ]
[ 4.157 2.4 ]
[ 0.693 3.6 ]
[ 1.386 4 ]
[ 2.078 3.6 ]
[ 2.771 4 ]
[ 3.464 3.6 ] ]
/edge [
[ 0 1 true ]
[ 1 2 true ]
[ 2 3 true ]
[ 3 4 true ]
[ 5 6 false ]
[ 6 7 true ]
[ 7 8 false ]
[ 8 9 false ]
[ 9 10 false ]
[ 10 11 true ]
[ 12 13 false ]
[ 13 14 false ]
[ 14 15 true ]
[ 15 16 true ]
[ 16 17 true ]
[ 17 18 true ]
[ 19 20 false ]
[ 20 21 false ]
[ 21 22 false ]
[ 22 23 false ]
[ 0 6 true ]
[ 2 8 false ]
[ 4 10 true ]
[ 5 12 false ]
[ 7 14 true ]
[ 9 16 false ]

```

Puzzle:



Solution:



```

[ 11 18 true ]
[ 13 19 false ]
[ 15 21 false ]
[ 17 23 false ] ]
/area [
[ 1.386 0.8 -1 0 ]
[ 2.771 0.8 3 0 ]
[ 0.693 2 2 0 ]
[ 2.078 2 -1 0 ]
[ 3.464 2 4 0 ]
[ 1.386 3.2 1 0 ]
[ 2.771 3.2 -1 0 ] ]
/style [
[ 1 0 0 1 0 0 ] ]
/area_outline [
[ 6 0 1 21 6 5 20 ]
[ 6 2 3 22 8 7 21 ]
[ 6 4 5 24 11 10 23 ]
[ 6 6 7 25 13 12 24 ]
[ 6 8 9 26 15 14 25 ]
[ 6 11 12 28 17 16 27 ]
[ 6 13 14 29 19 18 28 ] ]

```

The node-key describes the nodes of the graph. It's a list of pairs, describing the x- and y-coordinate of the node as floating point numbers.

The edge-key describes the edges of the node, as well as the solution of the puzzle. It's a list of three items per entry: The index of the node in the node-key, where the edge starts, the index of the node in the node-key, where the edge ends and a boolean value, telling if this edge is part of the solution or not.

The area-key is a list with four items per entry: The first two give x- and y-coordinate of the center of the area. The point defined by this coordinates has to be inside that area. It is often the balance point, but not necessary. The third entry is the value to be placed at that point, with -1 as a place holder for an empty areas. The last number gives the number of style to use when rendering this number. As some programs may rely upon it, all empty areas have to be given in this list.

The style-key is a list with six items per entry. They define an affine transformation, the first four define the matrix and the last two the translation vector. This transformation is used to describe the rendering style of that number.

An optional area_outline-key can be given. It is a list of polygons. The first entry of a polygon is the number of edges of that polygon. The other entries are indices to the list of edges. They have to be sorted to form a path around the area. The information provided by the area_outline-key can be deduced from

the values of the other keys. But as this is not an easy task to do, some programs might not work without this information.

A.16. Fillomino

There are two formats defined for the fillomino puzzle: Format 1 is a special format for rectangular grids. Format 2 is a more general format, that can handle arbitrary planar graphs.

A.16.1. Format 1

```

/X 4
/Y 4
/digits 2
/puzzle [
(+---+---+---+---+)
(|   1   3|)
(+  +  +  +  +)
(| 2     | |)
(+  +  +  +  +)
(| 1 4     |)
(+  +  +  +---+)
(|           2|)
(+---+---+---+---+) ]
/solution [
( 2 1 2 3)
( 2 4 2 3)
( 1 4 1 3)
( 4 4 2 2) ]

```

Puzzle:

	1		3
2			
1	4		
			2

Solution:

2	1	2	3
2	4	2	3
1	4	1	3
4	4	2	2

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2dx + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols. Edges can be marked by minus or pipe symbols. In the center of all cells, a number can be given. It has to be right aligned.

The solution-key is the same as the puzzle-key, but only the content of the cells is given.

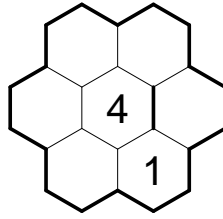
A.16.2. Format 2

```

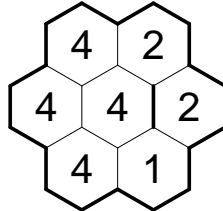
/edge [
[ 0 1 true ]
[ 1 2 false ]
[ 2 3 false ]
[ 3 4 false ]
[ 4 5 true ]
[ 5 0 true ]
[ 6 7 true ]
[ 7 8 true ]
[ 8 9 false ]
[ 9 2 false ]
[ 1 6 true ]
[ 3 10 false ]
[ 10 11 false ]
[ 11 12 true ]
[ 12 13 true ]
[ 13 4 true ]
[ 9 14 true ]
[ 14 15 false ]
[ 15 10 false ]
[ 8 16 true ]
[ 16 17 true ]
[ 17 18 true ]
[ 18 14 false ]
[ 15 19 false ]
[ 19 20 true ]
[ 20 21 true ]
[ 21 11 true ]
[ 18 22 true ]
[ 22 23 true ]
[ 23 19 true ] ]
/area [
[ 1.3856 0.8 -1 0 4 ]
[ 2.7712 0.8 1 0 1 ]
[ 0.6928 2 -1 0 4 ]
[ 2.0784 2 4 0 4 ]
[ 3.464 2 -1 0 2 ]
[ 1.3856 3.2 -1 0 4 ]
[ 2.7712 3.2 -1 0 2 ] ]
/style [
[ 1 0 0 1 0 0 ] ]
/node [
[ 1.3856 0 ]
[ 2.0784 0.4 ]
[ 2.0784 1.2 ]
[ 1.3856 1.6 ]
[ 0.6928 1.2 ]
[ 0.6928 0.4 ]
[ 2.7712 0 ]
[ 3.464 0.4 ]
[ 3.464 1.2 ]
[ 2.7712 1.6 ]

```

Puzzle:



Solution:



```

[ 1.3856 2.4 ]
[ 0.6928 2.8 ]
[ 0 2.4 ]
[ 0 1.6 ]
[ 2.7712 2.4 ]
[ 2.0784 2.8 ]
[ 4.1568 1.6 ]
[ 4.1568 2.4 ]
[ 3.464 2.8 ]
[ 2.0784 3.6 ]
[ 1.3856 4 ]
[ 0.6928 3.6 ]
[ 3.464 3.6 ]
[ 2.7712 4 ] ]
/area_outline [
[ 6 0 1 2 3 4 5 ]
[ 6 6 7 8 9 1 10 ]
[ 6 3 11 12 13 14 15 ]
[ 6 9 16 17 18 11 2 ]
[ 6 19 20 21 22 16 8 ]
[ 6 18 23 24 25 26 12 ]
[ 6 22 27 28 29 23 17 ] ]

```

The node-key describes the nodes of the graph. It's a list of pairs, describing the x- and y-coordinate of the node as floating point numbers.

The edge-key describes the edges of the node. It's a list of three items per entry: The index of the node in the node-key, where the edge starts, the index of the node in the node-key, where the edge ends and a boolean value which tells, if this edge should be a given border.

The area-key is a list with five items per entry: The first two give x- and y-coordinate of the center of the area. The point defined by this coordinates has to be inside that area. It is often the balance point, but not necessary. The third entry is the value to be placed at that point, with -1 as a place holder for empty areas. The fourth number gives the number of style to use when rendering this number. The last number is the solution. As some programs may rely upon it, all empty areas have to be given in this list.

The style-key is a list with six items per entry. They define an affine transformation, the first four define the matrix and the last two the translation vector. This transformation is used to describe the rendering style of that number.

An optional area_outline-key can be given. It is a list of polygons. The first entry of a polygon is the number of edges of that polygon. The other entries are indices to the list of edges. They have to be sorted to form a path around the area. The information provided by the area_outline-key can be deduced from

the values of the other keys. But as this is not an easy task to do, some programs might not work without this information.

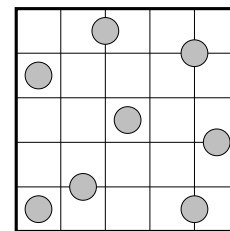
A.17. Galaxies

```

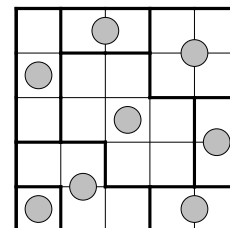
/X 5
/Y 5
/puzzle [
(+--+--+--++)
(|  0  |)
(+ + + + 0 +)
(|0      |)
(+ + + + + +)
(|  0  |)
(+ + + + +0+)
(|      |)
(+ +0+ + + +)
(|0      0 |)
(+--+--+--++) ]
/solution [
(+--+--+--++)
(| |  | |)
(+ +--+ + + +)
(| |  | |)
(+ + + +--++)
(| |  | |)
(+--+ + + +)
(|  |  | |)
(+--+ +--+--++)
(| |  | |)
(+--+--+--++) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

Vertices, edges and cells may contain a 0, to denote the center of a galaxy.

The solution-key is the same as the puzzle key. The borders of the areas are marked by minuses and pipes.

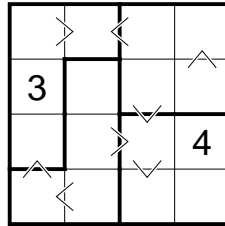
A.18. Greater-Than-Sudoku

```

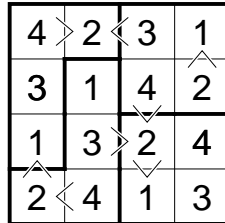
/size 4
/digits 2
/puzzle [
(+---+---+---+)
(| > C   |)
(+ +--+ + ^+)
(| 3| |   |)
(+ + +-V+---+)
(| | D   4|)
(+-A+ + v+ +)
(| < |   |)
(+---+---+---+) ]
/solution [
( 4 2 3 1)
( 3 1 4 2)
( 1 3 2 4)
( 2 4 1 3) ]

```

Puzzle:



Solution:



The size-key defines the size of the puzzle. It has to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2ds + 1$ and the height of the diagram is $2s + 1$, where s is the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

Inside the diagram, the edges between two areas are marked by pipes and minuses as well. In the cells, numbers can be given. If the digit-key is greater than 1, the numbers have to be right aligned.

Comparison symbols can be placed at the edges inside the puzzle. The symbols can be $<$, $>$, v and \wedge on edges that do not separate areas and C , D , V and A on edges that do separate areas. The symbols have to be centered in the edge. If digits is even, the place right to the center is used.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the cells is given.

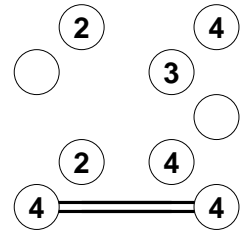
A.19. Hashi

```

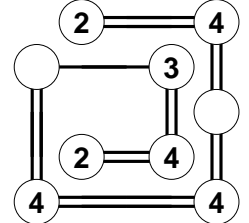
/X 5
/Y 5
/puzzle [
( 2 4)
(? 3 )
(   ?)
( 2 4)
(4===4) ]
/solution [
( 2==4)
(?--3H)
(H H?)
(H2=4H)
(4===4) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is x and the height of the diagram is y , where x and y are the size of the puzzle.

The symbols inside the puzzle can be a number from 1 to 8, which denote an island with this number inside it, a question mark, which denotes an island without known number, and bridges, which are marked by $-$, $=$, $|$, H .

The solution-key is the same as the puzzle-key. In the solution, question marks may be replaced by the correct number.

A.20. Heyawake

```

/X 4
/Y 4
/digits 2
/puzzle [
(+---+---+---+)
(| 2      | |)
(+---+---+---+)
(|          |--|)
(+---+---+---+)
(| | | 0| |)
(+ + + + +)
(|##| | | |)
(+---+---+---+) ]
/solution [
(# # )
(   )
( # )
(# #) ]

```

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $(d+1)x+1$ and the height of the diagram is $2y+1$, where x and y are the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

Inside the diagram, the edges between two areas are marked by pipes and minuses as well. Programs may assume, that all areas are rectangular. In the cells, numbers can be given. If the digit-key is greater than 1, the numbers have to be right aligned. Programs may assume, that there is only one number inside an area. The number should be placed at top left of the area, but programs must not rely on this. The cells can also be filled by hash keys as well as minus keys to denote a cell marked as filled or empty.

The solution-key has one symbol per cell. A black cell is written as # and non-black cells are written as space or dots.

Puzzle:

2			
			×
		0	

Solution:

			×
		0	

A.21. Hitori

```

/X 4
/Y 4
/digits 2
/puzzle [
( 9 6 6 )
( 9   9 )
(   12 )
(  12 )
( 3 312##) ]
/solution [
(..#. )
(#... )
(..#. )
(#..#) ]

```

Puzzle:

9	6	6	
9		9	
		12	
3	3	12	

Solution:

9	6		
		9	
	3	12	

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is dx and the height of the diagram is y , where x and y are the size of the puzzle and d is the value of the digits-key.

The cells may contain a number, which is placed at the rightmost position, a black cell, which is written by the hash symbol # or be empty. If the cell is black, all symbols should be hashes, but a program might only check one position.

The solution-key has one symbol per cell. A black cell is written as # and non-black cells are written as space or dots.

A.22. House-Tree-Animal

```

/X 5
/Y 5
/digits 2
/puzzle [
(
)
(
1 2 )
(
)
(
)
(
)
(
0 )
(
+-----)
(
2| # )
(
|* )
(
| )
(
1 | -)
(
| & ) ]
/solution [
(
&*&)
(*# *)
(& &#)
(# * )
(*& # ) ]

```

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

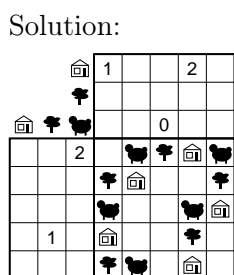
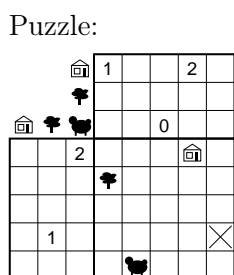
The puzzle-key gives the puzzle as ascii art. The width of the diagram is $x + 3d + 1$ and the height of the diagram is $y + 3d + 1$, where x is the width of the puzzle, y is the height of the puzzle and d is the value of the digits-key.

The place $3d \times 3d$ is marked with a plus. Right of that plus are minuses and below that plus are pipes. They primarily serve to help humans.

Inside the diagram, every cell of the puzzle contains one character, which can be # to denote a house, * to denote a tree, & to denote an animal and - to denote an empty cell.

Outside the diagram, the hints are given. At the left they are as many characters wide, as the digits-key gives. At the top they are as many characters high, as the digits-key gives. Numbers have to be right or bottom aligned.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the inner cells is given as single character.

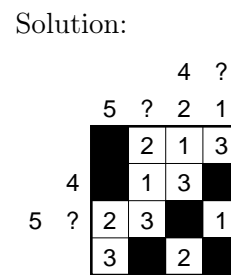
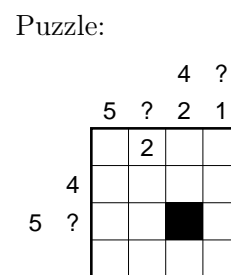


A.23. Japanese Sums

```

/X 4
/Y 4
/max 3
/left 2
/top 2
/puzzle [
(
4 ? )
(
5 ? 2 1 )
(
+-----)
(
|. 2 . . )
(
4| . . . . )
(
5 ?| . . # . )
(
|. . . . . ) ]
/solution [
(#213)
(#13#)
(23#1)
(3#2#) ]

```



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional max-key can be given, which denotes the maximum digit, that can appear in the puzzle. If omitted, 9 is assumed. The max-key cannot be greater than 9. The left-key and the top-key give the maximum number of numbers at the left and at the top of the puzzle. They should be given, as some programs might need them to know in advance the size of the puzzle-key.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x + 2l + 1$ and the height of the diagram is $2y + 2t + 1$, where x is the width of the puzzle, y is the height of the puzzle and l and t are the values of the left- and top-key.

The place $2l \times 2t$ is marked with a plus. Right of that plus are minuses and below that plus are pipes. They primarily serve to help humans, but the plus might be used by programs to decide upon the values of the left- and top-key if they are omitted. Minuses and pipes might be omitted.

Inside the diagram, every cell of the puzzle contains two characters, of which the left one should always be a space. The right one can be a space or a dot which are considered equivalent, but the dot should be used, as it helps humans. Furthermore, there can be a hash denoting a given black cell and digits from 1 to max.

Outside the diagram, the hints are given. They are always two cells wide. The content can be a number between 1 and $\frac{m(m+1)}{2}$, where m is the maximum digit given by the max-key, or a question mark, which denotes an unknown number. Rows or columns might

also be completely empty, which denotes, that there is no given information about this row or column.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the cells is given as single character.

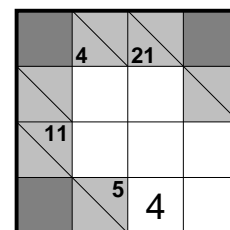
A.24. Kakuro

```

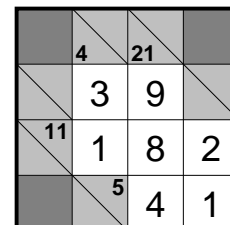
/X 4
/Y 4
/puzzle [
(#####)
(###421##)
(##  ##)
(##  ##)
(11  )
(##  )
(###54  )
(####  ) ]
/solution [
(####)
(#39#)
(#182)
(##41) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 2. In the size of the puzzle, the leftmost column and the topmost row, which both contain no empty cell, are included.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x$ and the height of the diagram is $2y$, where x is the width of the puzzle, y is the height of the puzzle.

The cells of the puzzle are given as blocks of size 2×2 . They can either be completely made of spaces, which denotes an empty cell. In that case, the top left place might contain a number to be displayed in this cell.

Or, they might denote clues. In that case the block is filled with hash signs. In the top two cells, the clue for the horizontal block can be given flushing to the right and in the bottom two cells, the clue for the vertical block can be given, again flushing to the right. Clues might be omitted.

The solution-key gives the solution as ascii art. The solution is made of hash keys which denote all non empty cells or digits.

A.25. Killer-Sudoku

```

/size 4
/puzzle [
(+--+--+--+ )
(|  |4  |)
(+ + + + +)
(|  |  |)
(+--+--+--+ )
(|1  |  |)
(+ + + + +)
(|  |  |)
(+--+--+--+ ) ]
/smallareas [
(+---+---+---+ )
(| 3  |## ##|)
(+---+---+---+ )
(| 5|##|  |##|)
(+ +---+ +---+ )
(|  |  | 4|)
(+---+---+---+ )
(| 7  |## ##|)
(+---+---+---+ ) ]
/solution [
(2143)
(4312)
(1234)
(3421) ]

```

Puzzle:

3		4	
5			
1			4
7			

Solution:

2	1	4	3
4	3	1	2
1	2	3	4
3	4	2	1

small area, have to be filled with two hash signs.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the cells is given.

The size-key defines the size of the puzzle. It has to be greater or equal to 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2s + 1$ and the height of the diagram is $2s + 1$, where s is the size of the puzzle.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

Inside the diagram, the edges between two areas are marked by pipes and minuses as well. In the cells, numbers can be given.

The smallareas-key gives the small areas as ascii art. The width of the diagram is $3s + 1$ and the height of the diagram is $2s + 1$, where s is the size of the puzzle.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

Inside the diagram, the edges of small areas are marked by pipes and minuses as well. In the cells, numbers can be given. If they are single digit, they have to be right aligned. It's possible to have small areas without number but each small area can only contain one number. Cells, that are not part of a

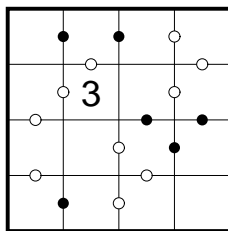
A.26. Kropki

```

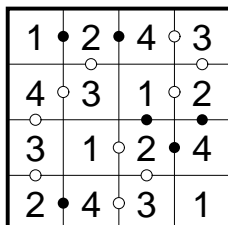
/size 4
/digits 2
/puzzle [
(+---+---+---+)
(| * * 0 |)
(+ + 0+ + 0+)
(| 0 3 0 |)
(+ 0+ + ** **+)
(| 0 * |)
(+ 0+ + 0+ +)
(| * 0 |)
(+---+---+---+) ]
/solution [
( 1 2 4 3)
( 4 3 1 2)
( 3 1 2 4)
( 2 4 3 1) ]

```

Puzzle:



Solution:



The size-key defines the size of the puzzle. It has to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2ds + 1$ and the height of the diagram is $2s + 1$, where s is the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

Dots can be placed at the edges inside the puzzle. Two types of dots are known: 0 for white dots and * for black dots. The dots have to be centered in the edge. If digits is even, the place right to the center is used.

In the cells, numbers can be given. If the digit-key is greater than 1, the numbers have to be right aligned.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the cells is given.

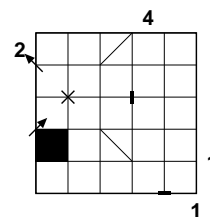
A.27. Laser

```

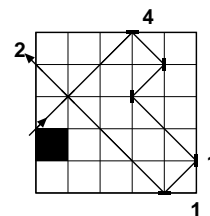
/X 5
/Y 5
/digits 2
/puzzle [
(
(
( 4
( +---+---+---+ )
( 2| / | )
( N + + + + )
( | | )
( + X + | + + )
( | | )
( 7 + + + + )
( |# N | )
( + + + + + 1)
( | | )
( +---+---+---+ )
(
( 1 ) ]
/solution [
( /N )
(N/ / )
(/N N )
( N N)
( N/) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers outside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x + 2d + 1$ and the height of the diagram is $2y + 2d + 1$, where x is the width of the puzzle, y is the height of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, starting at position $d \times d$, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram. The cells can contain an hash-key to indicate an unused cell. It can also contain a N or a slash-key to indicate a given piece of the laser beam. The plus symbols at the vertices can be replaced by an X to mark a crossing, by minus- or pipe-symbol to mark a mirror or by one of the symbols n, 7, N or / to mark the arrow entering (first two symbols) and leaving the diagram.

At the left and the top numbers can be given. At the right and the bottom also numbers can be given, but at the position of the grid lines. Numbers at the left and at the right have to be right aligned, while num-

bers at the top and at the bottom are written from top down and are bottom aligned.

The solution-key gives the solution as ascii art. The cells in the solution are made up of N and / to denote the laser beams or stay empty.

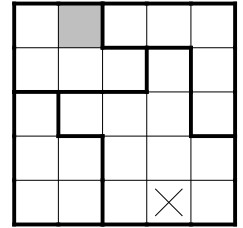
A.28. LITS

```

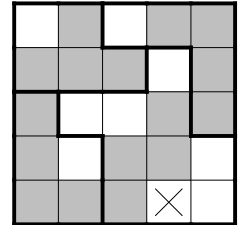
/X 5
/Y 5
/puzzle [
(+--+--+--++)
(| #| |)
(+ + +--+ +)
(| | | |)
(+--+--+ +)
(| | | |)
(+ +-+ + +-+)
(| | | |)
(+ + + + +)
(| | - |)
(+--+--+--++) ]
/solution [
(-#-##)
(###-#)
(#--##)
(#-##-)
(###--) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

Inside the diagram, the edges between two areas are marked by pipes and minuses as well. The cells can be filled by a hash key as well as a minus key to denote a cell marked as filled or empty.

The solution-key has one symbol per cell. A black cell is written as # and non-black cells are written as space or dots.

A.29. LITS plus

Same as LITS.

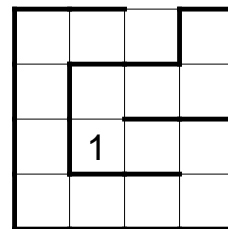
A.30. Magic Labyrinth

```

/size 4
/max 3
/puzzle [
(+--+ +--+ )
(|      | |)
(+ +--+ + )
(| |    |)
(+ + +--+ )
(| |1   |)
(+ +--+ + )
(|      |)
(+--+--+--+ ) ]
/solution [
(321 )
(1 23)
( 132)
(23 1) ]

```

Puzzle:



Solution:

3	2	1	
1		2	3
	1	3	2
2	3		1

The size-key defines the size of the puzzle. It has to be greater or equal to 1. The max-key gives the largest number to be inserted in the puzzle.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2s + 1$ and the height of the diagram is $2s + 1$, where s is the size of the puzzle.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram, except the one at the entrance of the labyrinth.

Inside the diagram, the edges that form the walls of the labyrinth are marked by pipes and minuses as well. In the cells, numbers can be given.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the cells is given.

A.31. Magnets

```

/X 5
/Y 5
/digits 2
/puzzle [
(   |           )
( + |         2 )
(   |           )
(  -|         1 )
(-----+--+--+--+ )
( 3 | | | | )
(   + +--+--+--+ )
(   | | | |#| )
(   +--+--+ + + )
( 1 1| | |#| |#| )
(   + + +--+--+ )
(   | | | | | )
(   +--+--+--+ + )
(   |- +| | | )
(   +--+--+--+ + ) ]
/solution [
(+--++)
(-##+#)
(##-#)
(##-##)
(-+--+ ) ]

```

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

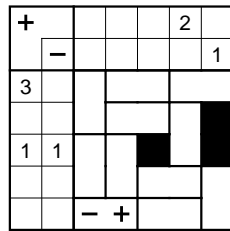
The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x + 2d + 1$ and the height of the diagram is $2y + 2d + 1$, where x and y are the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram. The top left corner is separated by minus and pipe symbols and a plus and a minus sign are placed in the corner. They are ment as help for human viewers only.

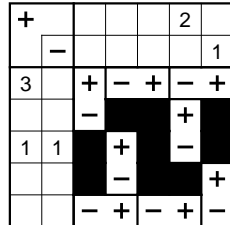
At the left and at the top numbers can be given. If d is greater than 1, the numbers at the top are given top down, that is, the decade is given above the unit position.

Inside the puzzle, the magnet plates are drawn using minus and pipes. All plates have to occupy two cells. There might be single cell plates, which all have to be neutral, that is, they have to be filled with a hash key #.

Puzzle:



Solution:



In the cells, +, - and # can be given, to indicate plus, minus and neutral cells. All cells of a plate have to be given. Programms might only check one of the two cells and calculate the other one.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the cells is given.

A.32. Masyu

```

/X 5
/Y 5
/puzzle [
(+--+--+--+--+ )
(| *  0  |)
(+ + + + +)
(|      - |)
(+ + + + +)
(|      # |)
(+ + + + +)
(|0      |)
(+ + + +|+ +)
(|      |)
(+--+--+--+--+ ) ]
/solution [
(+--+--+--+--+ )
(| *- -0- |)
(+ +|+ + +|+)
(|      - - |)
(+ +|+|+ + +)
(| -   # #|)
(+|+ +|+ + +)
(|0   -   |)
(+|+ + +|+ +)
(| - -0-   |)
(+--+--+--+--+ ) ]

```

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

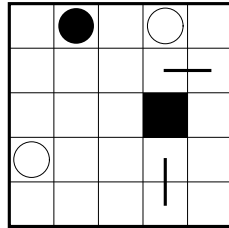
The cells may contain an 0 or a * to denote white or black circles, or a #, which is a cell, that might not be used by the path. Between two cells, there might be minus and pipe symbols to show part of the path.

The solution-key is the same as the puzzle-key.

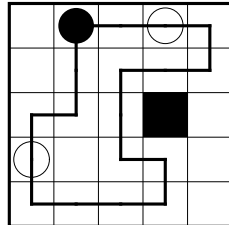
A.33. Narrow Cave

Same as Cave.

Puzzle:



Solution:



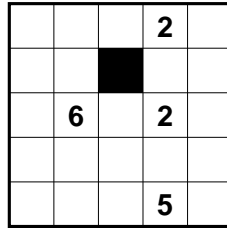
A.34. Nurikabe

```

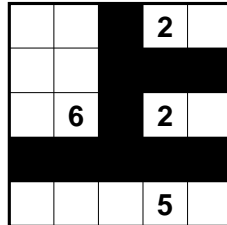
/X 5
/Y 5
/digits 2
/puzzle [
(   2   )
(   ##  )
(  6  2  )
(       )
(       5 ) ]
/solution [
( # )
( ###)
( # )
(#####)
(   ) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is dx and the height of the diagram is y , where x and y are the size of the puzzle and d is the value of the digits-key.

Cells can contain # to denote part of the walls or a number, which has to be right aligned.

The solution-key has one symbol per cell. The hash-symbol for walls and a space else.

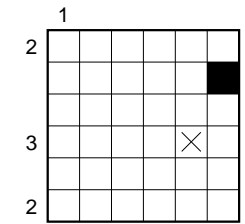
A.35. Pentomino Search

```

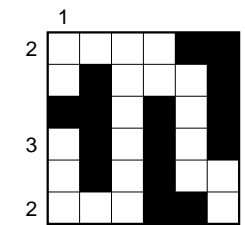
/X 6
/Y 6
/used (LLY)
/digits 2
/puzzle [
(       )
(  1   )
( +-----)
( 2|   )
( |   #)
( |   )
( 3|  - )
( |   )
( 2|   ) ]
/solution [
(   ##)
( #  #)
(## # #)
( # # #)
( # # )
(   ## ) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. The used-key gives a string with the pentominos used. It consists out of letters F,I,L,N,P,T,U,V,W,X,Y and Z. If a letter is given several times, the pentomino has to be placed that amount of times in the diagram. The used-key can be omitted. In that case, the set of the pentominos is unknown. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $x + d + 1$ and the height of the diagram is $y + d + 1$, where x and y are the size of the puzzle and d is the maximum number of digits.

Each cell can be completely filled with hash-symbols to denote a block, completely filled with minus-symbols to denote an empty cell or be blank. At the left and the top numbers can be given. The numbers at the left have to be right aligned, the number at the top have to be bottom aligned.

The solution-key is the same as the puzzle-key, but only the content of the cells is given.

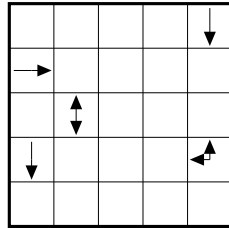
A.36. Pentopia

```

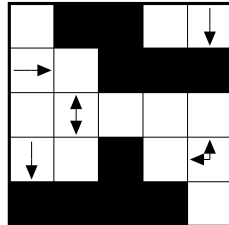
/X 5
/Y 5
/puzzle [
(   v)
(>  )
( |  )
(v  S)
(    ) ]
/solution [
( ## )
( ###)
(    )
( #  )
(#### ) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is x and the height of the diagram is y , where x and y are the width and the height of the puzzle.

Each cell can contain various symbols. See the table at the end of this section for more details.

The solution-key is the same as the puzzle-key. Arrows can be omitted.

- > right arrow
- < left arrow
- ^ top arrow
- v bottom arrow
- left-right arrow
- | top-bottom arrow
- Q right-bottom arrow
- W left-bottom arrow
- A right-top arrow
- S left-top-arrow
- T T-shape arrow down
- F T-shape arrow right
- G T-shape arrow left
- V T-shape arrow up
- + arrow in all four directions
- # black cell
- X white cell

A.37. Pills

```

/X 4
/Y 4
/max 3
/digits 2
/puzzle [
(   2 2 )
(  1 2 1 1)
( 4 0 1 2 1)
(  1 1 2 0)
( 1 0 1 0 2) ]
/solution [
(   )
( - |)
(   )
( - ) ]

```

Puzzle:

		2	2	
	1	2	1	1
4	0	1	2	1
	1	1	2	0
1	0	1	0	2

Solution:

		2	2	
	1	2	1	1
4	0	1	2	1
	1	1	2	0
1	0	1	0	2

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. The max-key gives the largest value of a pill. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 2.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $d(x + 1)$ and the height of the diagram is $y + 1$, where x and y are the size of the puzzle and d is the value of the digits-key.

Each cell must contain a number, which is right aligned. At the left and the top numbers can be given. They also have to be right aligned.

The solution-key has one symbol per cell. A pipe symbol for the center of a vertical cell and a minus sign for the center of an horizontal cell.

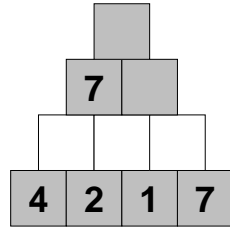
A.38. Pyramid

```

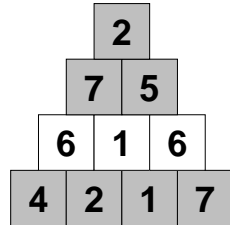
/size 4
/puzzle [
(# . . )
(# 7 . . )
(= . . . )
(#4 2 1 7) ]
/solution [
( 2 )
( 7 5 )
( 6 1 6 )
(4 2 1 7) ]

```

Puzzle:



Solution:



The size-key defines the size of the puzzle. It has to be greater or equal to 2.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2s$ and the height of the diagram is s , where s is the size of the puzzle.

The leftmost symbol in every row must be = (a white row) or # (a grey row). In the rest of the grid, the cells of the pyramid are given to form a pyramid. An empty cell can be either a space symbol or a dot.

The solution is the same as the puzzle grid, with one exception: The leftmost column is missing.

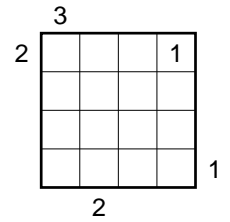
A.39. Skyscrapers

```

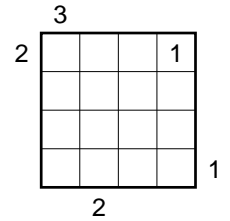
/size 4
/digits 2
/puzzle [
( 3 . . . )
( 2 . . 1 . )
( . . . . )
( . . . . )
( . . . . 1)
( . . . . 2) ]
/solution [
( 2 4 3 1)
( 3 1 4 2)
( 4 2 1 3)
( 1 3 2 4) ]

```

Puzzle:



Solution:



The size-key defines the size of the puzzle. It has to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width and height of the diagram are $d(s+2)$, where s is the size of the puzzle and d is the maximum number of digits.

The numbers at the border are given in the leftmost, rightmost column and top and bottom row. In the center, numbers can be given. All numbers are right aligned.

The solution-key gives the solution as ascii art. In the solution diagram, only the center is given.

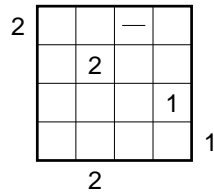
A.40. Skyscrapers with Parks

```

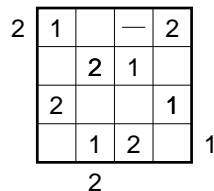
/size 4
/parks 2
/digits 2
/puzzle [
(          )
( 2      - )
(    2    )
(          1 )
(            1)
(    2    ) ]
/solution [
( 1      2)
(  2 1  )
( 2     1)
(  1 2  ) ]

```

Puzzle:



Solution:



The size-key defines the size of the puzzle. It has to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1. The parks-key gives the number of parks in each row and column.

The puzzle-key gives the puzzle as ascii art. The width and height of the diagram are $d(s+2)$, where s is the size of the puzzle and d is the maximum number of digits.

The numbers at the border are given in the leftmost, rightmost column and top and bottom row. In the center, numbers or a minus sign can be given. All numbers and signs are right aligned.

The solution-key gives the solution as ascii art. In the solution diagram, only the center is given.

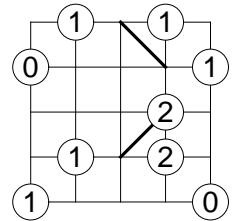
A.41. Slalom

```

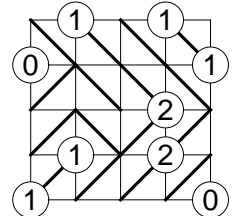
/X 4
/Y 4
/puzzle [
(+ 1 + 1 +)
(      N   )
(0 + + + 1)
(          )
(+ + + 2 +)
(      /   )
(+ 1 + 2 +)
(          )
(1 + + + 0) ]
/solution [
(NNNN)
(/NNN)
(/N//)
(///// ) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x+1$ and the height of the diagram is $2y+1$, where x and y are the size of the puzzle.

Vertices can be numbers from 0 to 4 or a plus symbol to denote an empty vertice.

The cells may contain a slash or an N which serves as a replacement for the backslash.

The solution-key is the same as the puzzle-key, but only the content of the cells is given.

A.42. Snake

```

/X 5
/Y 5
/max 11
/digits 3
/puzzle [
(           1   )
(           )
(    ###  7   )
(           )
(  3         )
(   2        ) ]
/solution [
(           )
(    ###  7  8  9)
(         6   10)
(  1     5   11)
(  2  3  4     ) ]

```

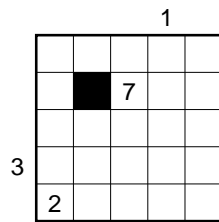
The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. The max-key gives the length of the snake. It can be omitted. In that case, the length of the snake is unknown. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 2.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $d(x + 1)$ and the height of the diagram is $y + 1$, where x and y are the size of the puzzle and d is the value of the digits-key.

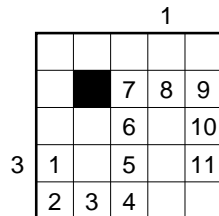
Each cell can contain a number, which is right aligned, be completely filled with hash-symbols to denote a cell, that is not used by the snake, or be blank. At the left and the top numbers can be given. They also have to be right aligned.

The solution-key is the same as the puzzle-key, but the leftmost column and the topmost row are removed.

Puzzle:



Solution:



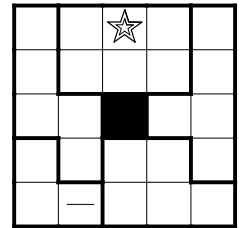
A.43. Starbattle

```

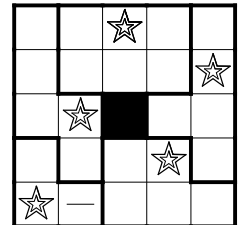
/stars 1
/size 5
/puzzle [
(+--+--+--++)
(| | * | |)
(+ + + + +)
(| |   | |)
(+ +-+--+--++)
(|  #|  |)
(+--+--+--++)
(| | |  |)
(+ +-+ + +-+)
(| -|   |)
(+--+--+--++) ]
/solution [
( * )
(   )
( * )
( * )
( * ) ]

```

Puzzle:



Solution:



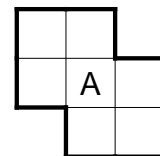
The size-key defines the size of the puzzle. It has to be greater or equal to 1. An optional stars-key can be given, which denotes the number of stars used in the puzzle. If omitted 2 is assumed.

The puzzle-key gives the puzzle as ascii art. The width and height of the diagram are $2x + 1$, where x is the size of the puzzle.

All vertices are marked as plus symbols, the edges at the border and between areas are marked by minus and pipe symbols.

The cells may contain *, -, # or be empty. A star-symbol marks a star, that is already given. A minus marks a cell, that cannot contain a star. The hash-sign is used to mark cells, that do not belong to any area.

To understand the difference between - and # is important. Consider the following diagram:



If a hash-symbol is placed inside the cell marked with an A the area would be separated in two areas, while a minus sign would not separate the area.

The solution-key gives the solution as ascii art. In the solution diagram, only the stars are given.

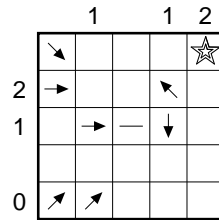
A.44. Starry Sky

```

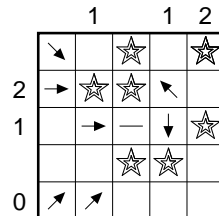
/X 5
/Y 5
/digits 1
/puzzle [
( 1 12)
( c *)
(2d q )
(1 d-x )
( )
(0ee ) ]
/solution [
( * *)
( ** )
( *)
( ** )
( ) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $d(x + 1)$ and the height of the diagram is $y + 1$, where x and y are the size of the puzzle and d is the maximum number of digits.

The content of a cell of the puzzle-key can either be a number, which has to be right aligned, or a symbol, which is placed at the rightmost place of the cell. The symbol can be a star or a minus. Letters are used to symbolize arrows. The letters q, w, e, a, d, y, x and c are used to give arrows that point away from the s on a keyboard. Note, that on a german keyboard y is in place of a z on several other keyboards...

The solution-key has one symbol per cell. A star is written as *.

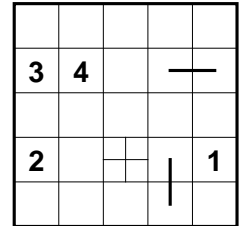
A.45. Stations

```

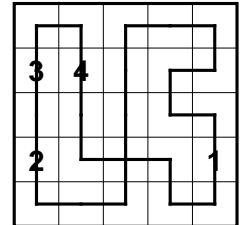
/X 5
/Y 5
/digits 2
/puzzle [
(+---+---+---+---+)
(| - - - - |)
(+ | + | + | + | +)
(| 3 4 - |)
(+ | + | + | + | +)
(| - - - - |)
(+ | + | + | + | +)
(| 2 - + - 1 |)
(+ | + | + | + | +)
(| - - - - |)
(+---+---+---+---+) ]
/solution [
(+---+---+---+---+)
(| - - - - |)
(+ | + | + | + | +)
(| 3 4 - |)
(+ | + | + | + | +)
(| - - - - |)
(+ | + | + | + | +)
(| 2 - + - 1 |)
(+ | + | + | + | +)
(| - - - - |)
(+---+---+---+---+) ]

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $(d + 1)x + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

The cells are d characters wide. The cell can contain the symbol + which is a crossing and has to be centered. If digits is even, the place right to the center is used. The cell can also contain a number, which has to be right aligned.

Between two cells, there might be minus and pipe symbols to show part of the path.

The solution-key is the same as the puzzle-key.

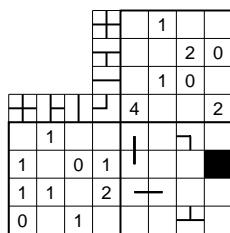
A.46. Subway

```

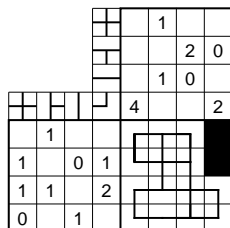
/X 4
/Y 4
/digits 2
/puzzle [
(
)
(      1  )
(
)
(      2 0)
(
)
(      1 0 )
(
)
(      4   2)
(  1   . . W .)
(      |   )
(  1   0 1. . . #)
(
)
(  1 1   2.-. . .)
(
)
(  0   1   . . V .) ]
/solution [
(..-. #)
(| | | )
(..-. #)
( | | )
(..-.-.)
(| | |)
(..-.-.) ]

```

Puzzle:



Solution:



the solution diagram, only dots and hashes are used in the cells and pipes and minues are used to give lines.

- horizontal line
- | vertical line
- Q top-left corner
- W top-right corner
- A bottom-left corner
- S bottom-right corner
- T T-shape down
- F T-shape right
- G T-shape left
- V T-shape up
- + crossing

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x + 4d - 1$ and the height of the diagram is $2y + 4d - 1$, where x is the width of the puzzle, y is the height of the puzzle, d is the value of the digits-key.

The places $2i + 4d \times 2j + 4d$ mark the cells. They are marked with a dot. The dot can be replaced with a hash to mark a blackend cell. Also, various letters and symbols can be used to show a given content of a cell. See table at the end of the specification.

Between two cells a minus or a pipe can be placed to denote a given line.

Outside the diagram, the hints are given. At the left they are as many charakters wide, as the digits-key gives. At the top they are as many charakters high, as the digits-key gives. Numbers have to be right or bottom aligned.

The solution-key gives the solution as ascii art. In

A.47. Sudoku

```

/size 4
/digits 2
/puzzle [
(+---+---+---+)
(|   |   |   |
(+  +---+  +  +)
(| 1| |   |   |
(+  +  +---+---+)
(| | | 2 |   |
(+---+  +  +  +)
(|   3| 4 |   |
(+---+---+---+) ]
/solution [
( 3 2 1 4)
( 1 4 3 2)
( 4 1 2 3)
( 2 3 4 1) ]

```

Puzzle:

1			
		2	
	3	4	

Solution:

3	2	1	4
1	4	3	2
4	1	2	3
2	3	4	1

The size-key defines the size of the puzzle. It has to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2ds + 1$ and the height of the diagram is $2s + 1$, where s is the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

Inside the diagram, the edges between two areas are marked by pipes and minuses as well. In the cells, numbers can be given. If the digit-key is greater than 1, the numbers have to be right aligned.

The solution-key gives the solution as ascii art. In the solution diagram, only the content of the cells is given.

A.48. Tapa

```

/X 6
/Y 6
/puzzle [
(
)
(
)
( 11 51 )
( 11 )
(## )
(## )
( 22 )
( 1 )
( 7 )
(
)
(2 -- )
( -- ) ] ]
/solution [
(## ###)
(# #)
(## # #)
(# ###)
(### #)
( #####) ]

```

Puzzle:

		1	1	5	1
		2	2	1	
					7
2	-				

Solution:

		1	1	5	1
		2	2	1	
					7
2	-				

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x$ and the height of the diagram is $2y$, where x and y are the size of the puzzle.

The cells of the puzzle are given as blocks of size 2×2 , that contain the clues. The first clue has to be placed top left, the next clue top right, than bottom left and the last clue at bottom right. They should be in descending order for better readability, but programmes must not rely on that.

Cells can also be filled with # or -, denoting an all-ready filled and a cell marked empty. If this symbols are used, all four places should be filled with it, but programs may rely only on the top left clue.

The solution-key gives the solution as ascii art. In the solution diagram, only the hashes are given to denote the cells, that have to be blackend.

A.49. Tents

```

/X 5
/Y 5
/digits 2
/puzzle [
(      )
(      11 )
( +-----)
( | * * )
( |      )
( |      )
( 0|* * )
( 2|  * ) ]
/solution [
( A   )
(   A )
( A   )
(     )
( A A ) ]

```

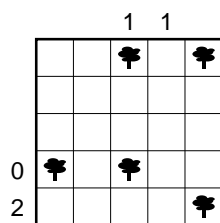
The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $x + d + 1$ and the height of the diagram is $y + d + 1$, where x and y are the size of the puzzle and d is the maximum number of digits.

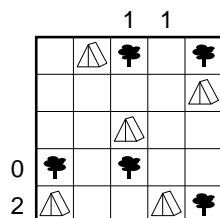
Each cell can be * to denote a tree or empty. At the left and the top numbers can be given. The numbers at the left have to be right aligned, the number at the top have to be bottom aligned.

The solution-key gives the solution as ascii art. The width of the diagram is x and the height of the diagram is y . Tents are marked as A.

Puzzle:



Solution:



A.50. Thermometers

```

/X 5
/Y 5
/digits 2
/puzzle [
(      )
(      1 1 3 3 )
( +-----)
( 1|----0 --0 )
( |      )
( 2|-----0 0 )
( |      | )
( 3|--0 0-- | )
( |      )
( || 0-- 0-- )
( ||      )
( 3|0 0-----) ]
/solution [
( #   )
( ##  )
( ### )
( ##  )
(### ) ]

```

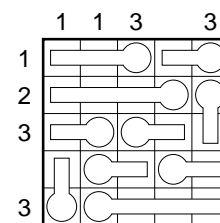
The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x + d + 1$ and the height of the diagram is $2y + d + 1$, where x and y are the size of the puzzle and d is the maximum number of digits.

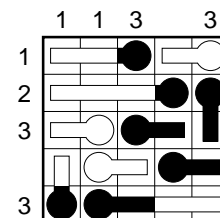
Thermometers are given as an 0, which continues with minuses or pipes. At the left and the top numbers can be given. The numbers at the left have to be right aligned, the number at the top have to be bottom aligned.

The solution-key gives the solution as ascii art. The width of the diagram is x and the height of the diagram is y . Blackend parts of the thermometers are marked by #.

Puzzle:



Solution:



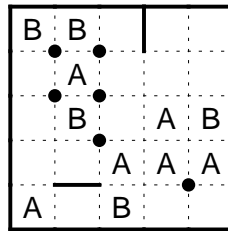
A.51. Yagit

```

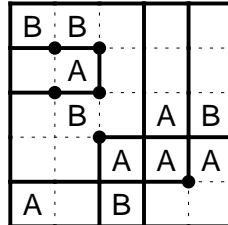
/X 5
/Y 5
/puzzle [
(+--+--+--+--+
(|B B | |)
(+ * * + + +)
(| A | |)
(+ * * + + +)
(| B A B|)
(+ + * + + +)
(| A A A|)
(+ +-+ + * +)
(|A B | |)
(+--+--+--+--+ )
/solution [
(+--+--+--+--+
(| | | |)
(+--+ + + +)
(| | | |)
(+--+ + + +)
(| | | |)
(+ + +--+--+)
(| | | |)
(+--+--+--+ +)
(| | | |)
(+--+--+--+--+ )

```

Puzzle:



Solution:



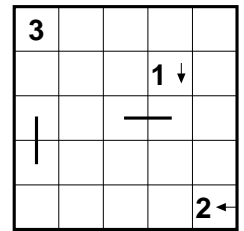
A.52. Yajilin

```

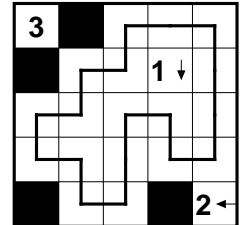
/X 5
/Y 5
/digits 1
/puzzle [
(+--+--+--+--+
(|3 | | |)
(+ + + + +)
(| 1x |)
(+ + + + +)
(| - |)
(+ |+ + + +)
(| |)
(+ + + + +)
(| 2a|)
(+--+--+--+--+ )
/solution [
(+--+--+--+--+
(| # - - |)
(+ + +|+ +|+)
(|# - |)
(+ +|+ + +|+)
(| - - |)
(+|+ +|+|+)
(| - - |)
(+ +|+ + +)
(|# - # |)
(+--+--+--+--+ )

```

Puzzle:



Solution:



The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $2x + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle.

The edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram. Edges with a dot are marked by * and all other edges are marked by a plus symbol.

Cells can contain A, B or be empty.

Between two cells, there might be minus and pipe symbols to show part of the solution.

The solution-key gives the puzzle as ascii art. And is identical to the puzzle-key.

The X-key and the Y-key define the size of the puzzle. Both have to be greater or equal to 1. An optional digits-key can be given, which denotes the maximum number of digits of numbers inside the diagram. If omitted, it is assumed to be 1.

The puzzle-key gives the puzzle as ascii art. The width of the diagram is $(d + 2)x + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle and d is the maximum number of digits.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

The cells are $d + 1$ characters wide. All, but the rightmost character can be a number or empty. If it is a number, it has to be right aligned and the rightmost character gives the direction of the arrow: The letters a, w, d and x can be used to give arrows that point away from the s on a keyboard. The character can also be empty. In that case the number is the sum of all four directions (total yajilin).

Between two cells, there might be minus and pipe symbols to show part of the path.

The solution-key gives the puzzle as ascii art. The

width of the diagram is $2x + 1$ and the height of the diagram is $2y + 1$, where x and y are the size of the puzzle.

All vertices are marked as plus symbols, the edges at the border are marked by minus and pipe symbols to indicate the border of the puzzle diagram.

The cells are one character wide. They contain a hash symbol, if the cell is black.

Between two cells, there are minus and pipe symbols to show the solution.