1. Programming Task: **Killer Sudoku**

Write a program to implement a constraint solver to solve a Killer Sudoku problem.

Requirements

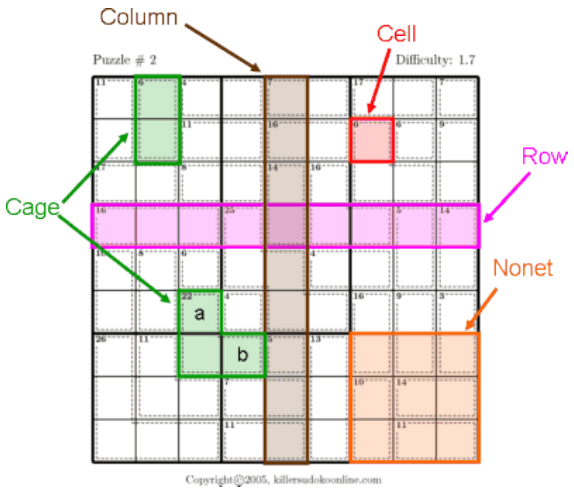
Programming Language: C++
Constraint Programming Library: Gecode 6.2.0[i] (available at https://www.gecode.org/)
Hint: Use all-different constraint `distinct` in the model of the problem
Execution Platform: Linux

Killer Sudoku Rules[ii]

Summary

The rules for Killer Sudoku are simple. The rules for regular sudoku apply, with one additional rule: The sum of the cells in a cage must equal the total given for the cage. Each digit in the cage must be unique.

Terminology



Cell	A single box. It can contain any digit from one to nine.
Row	A horizontal group of nine cells. Each cell in the row must contain a different digit.
Column	A vertical group of nine cells. Each cell in the column must contain a different digit.

Nonet	A 3x3 grid of cells. It is surrounded by bold lines. Each cell in the nonent must contain a different digit.
Cage	A group of cells surrounded by a dashed line. The values of the cells in a cage must sum up to the total specified in the upper right of the cage. The values in a cage must be unique even if the rules of regular sudoku would allow for duplicates. For example, the cells labeled 'a' and 'b' in the figure cannot have the same value.
Region	An area of the puzzle that cannot contain duplicate digits. That is, a row, a column, a nonet, or a cage.

Rules

- As in regular sudoku, every cell in each row, column, and nonet must contain a unique digit. In other words, each row, column, and nonet must contain all the digits from one to nine.
- The values of the cells a cage must sum up to the total for that cage.
- The values of the cells in a cage must be unique.

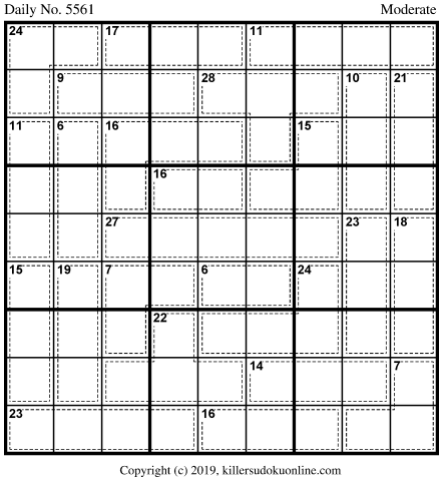
Input, Execution, Output

Text File Input

Each input file corresponds to a problem instance. Each problem instance contains a set of cages and their total. A cell is specified as a pair of row index and column index such that the top left most cell is referred as (1, 1). Each line in the input file specifies the cells in a cage and the total for that cage in the following format:

[total of the cage] [number of cells] [row index of cell 1] [column index of cell 1] [row index of cell 2] [column index of cell 2] … [row index of cell n] [column index of cell n]

Example:



The followings are content of the input file for the above problem.

```
24 3 1 1 1 2 2 1
17 3 1 3 1 4 1 5
11 4 1 6 1 7 1 8 1 9
9 3 2 2 2 3 2 4
28 4 2 5 2 6 2 7 3 6
10 3 2 8 3 8 4 8
21 3 2 9 3 9 4 9
11 3 3 1 4 1 5 1
6 3 3 2 4 2 5 2
16 4 3 3 3 4 3 5 4 3
15 3 3 7 4 6 4 7
16 2 4 4 4 5
27 5 5 3 5 4 5 5 5 6 5 7
23 3 5 8 6 8 7 8
18 3 5 9 6 9 7 9
15 3 6 1 7 1 8 1
19 3 6 2 7 2 8 2
7 3 6 3 6 4 7 3
6 2 6 5 6 6
24 4 6 7 7 5 7 6 7 7
22 4 7 4 8 3 8 4 8 5
14 3 8 6 8 7 8 8
7 3 8 9 9 8 9 9
23 4 9 1 9 2 9 3 9 4
16 3 9 5 9 6 9 7
```

Execution and Output

The program will be run with the text file input as a command line argument:

```
$ ./killer_sudoku cages.txt
```

The program should print out the solution in 9 x 9 format, each line contains 9 numbers, printed in 9 lines:

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

Submission

Name the program source file as `killer_sudoku.cpp` and submit the program source.

End

[i] Compiling and installing Gecode: https://www.gecode.org/doc/2.2.0/reference/PageComp.html
[ii] Killer Sudoku Rules: https://www.killersudokuonline.com/rules.html