Problem 1 - Killer Sudoku (Programming) (30 points)

Problem Description

Today, boook and ltf found an interesting game. It is called `killer sudoku`. The game goes as follows:

- There's a 9×9 board containing 81 grids. Like in the original sudoku, you need to fill an integer in each grid. The integers you fill must be in the range of $[1, 9]$.
- Every grid has a color assigned to it, and there are at most 81 colors. The grids that have the same color are connected.
- The integers filled in the grids need to satisfy the following requirements:
  - In each row, each number must appear exactly once.
  - In each column, each number must appear exactly once.
  - The board is divided into nine 3×3 blocks, each block contains 9 grids. In each block, each number must appear exactly once.
  - For each color, the number written on those colored grids must appear at most once.
  - For each color, the summation of the numbers written on those colored grids must equal to a target value.

Now, given the color on each grid and the target value of each color, please help boook and ltf find a way to complete the killer sudoku.

The colors in the following figures are just for separating different components, they do not represent the colors mentioned in the problem description.

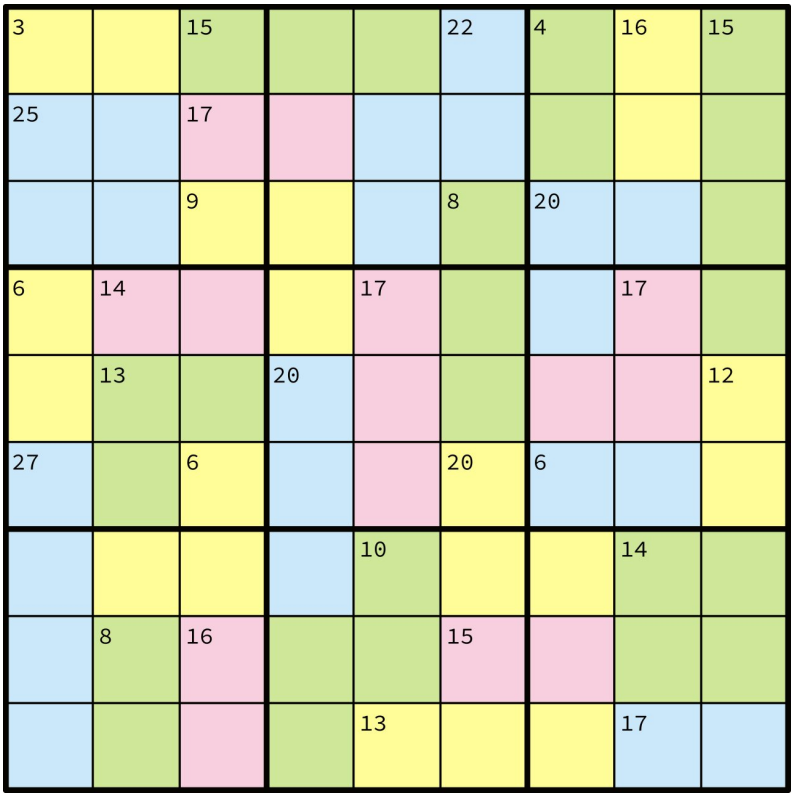

Figure 1. One killer sudoku (source: wiki)

Figure 2. The solution of a killer sudoku (source: wiki)

Input

In the first 9 lines, each line contains a string with 9 characters, representing a 9×9 board with colors assigned. Each character represents a specific color. It is guaranteed that the grids having the same color are connected, and the ASCII values of all characters are ranged in $[33, 126]$.

Let c be the number of different colors assigned on the board. In the following c lines, each line contains a character a and a positive integer k, meaning that the color represented by a has a target value k. It is guaranteed that all of these c characters appear on the board and are distinct.

Test Group 0 (0 %)

- Sample Input

Test Group 1 (10 %)

- $c = 9$

Test Group 2 (20 %)

- $c \geq 75$

Test Group 3 (40 %)

- $c \geq 37$

Test Group 4 (30 %)

- No additional constraints.

Output

Print a 9×9 board filled with integers satisfying all constraints mentioned above. If there are multiple solutions, you may print any one of them. It is guaranteed that there exists a solution satisfying all constraints.

Sample Input 1

```
AABBBCDEF
GGHHCCDEF
GGIICJKKF
LMMINJKOF
LPPQNJOOR
SPTQNUVVR
STTQWUUXX
SYZWW@@XX
SYZW###$$
A 3
B 15
C 22
D 4
E 16
F 15
G 25
H 17
I 9
J 8
K 20
L 6
M 14
N 17
O 17
P 13
Q 20
R 12
S 27
T 6
U 20
V 6
W 10
X 14
Y 8
Z 16
@ 15
# 13
$ 17
```

Sample Output 1

```
215647398
368952174
794381652
586274931
142593867
973816425
821739546
659428713
437165289
```

Hints

1. In the following we introduce the *exact cover problem*, which may help you to solve the problem. Formally, given a set $X = \{x_1, x_2, \ldots, x_n\}$ with n elements, and several subsets Y_1, Y_2, ..., Y_m of X. You are asked to determined whether it is possible to choose some indexes $i_1, i_2, \ldots, i_k$ such that

$$X = \bigsqcup_{j=1}^{k} Y_{i_j}.$$

That is, X is the disjoint union of subsets $Y_{i_1}, Y_{i_2}, \ldots, Y_{i_k}$.

You are given a header file `helper.h` which is helpful to solve the *exact cover problem*.

- `void DLX::Init(int n)`: Initialization, the set X is set to be size n.
- `void DLX::AddRow(int rr, vector<int> &sol)`: Insert a subset `sol`, regarded as $Y_{rr}(\forall x_i \in Y_{rr}, 1 \leq x_i \leq n)$, of X. Note that `sol` should be sorted in increasing order. Note that the subset **can not be empty**
- `vector<int> DLX::Solver()`: Return the vector of chosen indexes if the answer exists, or an empty vector otherwise.