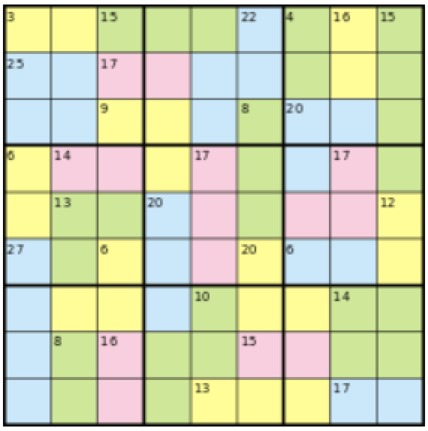
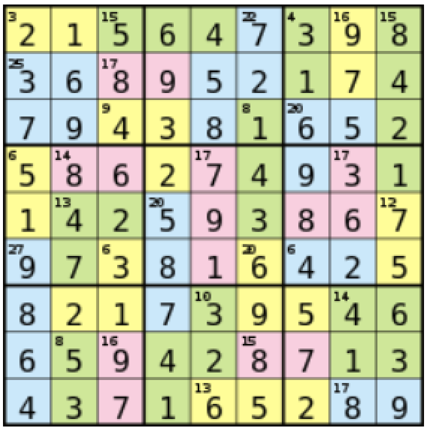
Description

Sumdoku is a variant of classic sudoku. In classic sudoku, the goal is to fill a 9×9 grid such that for each row, column and box contains all digits form 1 to 9. Sumdoku introduces a new rule - "cage". Several cells will be grouped together in a cage, and each cage is attached with a number. The sum of digits in each cage must meet the attached number.

Below is an example of sumdoku. A cage is denoted by connected cells with the same color. There is a number attached at the top-left corner of each cage





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

Given a sumdoku where all grids are already filled, your task is to verify whether it violates any rule or not.

Input

The first line of the input contains an integer T — the number of testcases.

The first line in each testcase contains an integer n, being the number of cages.

The second line in each testcase contains n integers $c_1, c_2, \cdots, c_n$, where c_i is the number attached to cage i.

The following nine lines in each testcase contains a 9×9 matrix M which specifies the cages. $M_{i,j}$ is the cage cell (i, j) belongs to.

The final nine lines in each testcase contains a 9×9 grid which is the given filled sumdoku.

Restrictions

- $1 \leq T \leq 30$
- $1 \leq n \leq 81$
- $1 \leq c_i \leq 500$
- $1 \leq M_{i,j} \leq n$

Output

Please output "YES" (without quotation mark) if the given filled sumdoku is valid. Otherwise output "NO" (without quotation mark).

Sample Input 1

```
2
29
3 15 22 4 16 15 25 17 9 8 20 6 14 17 17 1
```

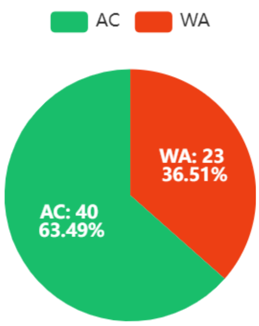
Sample Output 1

```
YES
NO
```

Information

ID	sumdol
Time Limit	1000M
Memory Limit	256M
IO Mode	Standard I
Created By	ta_redle
Level	Hidde
Score	1(
Tags	Sho

Statistic Deta



```
12 13 13 9 14 10 11 15 6
12 16 16 17 14 10 15 15 18
19 16 20 17 14 21 22 22 18
19 20 20 17 23 21 21 24 24
19 25 26 23 23 27 27 24 24
19 25 26 23 28 28 28 29 29
2 1 5 6 4 7 3 9 8
3 6 8 9 5 2 1 7 4
7 9 4 3 8 1 6 5 2
5 8 6 2 7 4 9 3 1
1 4 2 5 9 3 8 6 7
9 7 3 8 1 6 4 2 5
8 2 1 7 3 9 5 4 6
6 5 9 4 2 8 7 1 3
4 3 7 1 6 5 2 8 9
1
487
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
9 9 9 9 9 9 9 9 9
8 5 5 5 6 6 6 6 6
8 5 3 3 3 3 3 3 6
8 5 3 1 1 1 2 3 6
8 5 3 1 1 1 2 4 6
8 5 2 1 1 1 2 4 6
8 5 2 2 2 2 2 4 7
8 5 4 4 4 4 4 4 7
8 8 7 7 7 7 7 7 7
```

Hint

Explanation of Sample IO

This input contains $T = 2$ testcases.

The first testcase contains the filled sumdoku shown in the figure of the problem description, which is a valid sumdoku.

The second testcase contains another filled sumdoku, which is obvious that it violates multiple rules.

Detailed Restrictions

For test ID 1 (5% of total points):

- It is identical to sample input

For test ID $2, 3$ (10% of total points):

- $T = 1$

For test ID 4 (5% of total points):

- It doesn't violate the rule of classic sudoko. (i.e. For the given martix M, each row, each column, and each box contains numbers from 1 to 9.)

For test ID 5 (5% of total points):

For test ID $6, 7, 8, 9, 10$ (75% of total points)

- No additional restrictions in these tests.

.anguage: C++

Theme: Solarized Light

✓ You have solved the problem

Submit for Sample Test

Submit

Sample Test Input

Sample Test Output

```
29
3 15 22 4 16 15 25 17 9 8 20 6 14 17 17
13 20 12 27 6 20 6 10 14 8 16 15 13 17
1 1 2 2 2 3 4 5 6
7 7 8 8 3 3 4 5 6
7 7 9 9 3 10 11 11 6
12 13 13 9 14 10 11 15 6
12 16 16 17 14 10 15 15 18
19 16 20 17 14 21 22 22 18
19 20 20 17 23 21 21 24 24
19 25 26 23 23 27 27 24 24
19 25 26 23 28 28 28 29 29
2 1 5 6 4 7 3 9 8
3 6 8 9 5 2 1 7 4
7 9 4 3 8 1 6 5 2
5 8 6 2 7 4 9 3 1
1 4 2 5 9 3 8 6 7
9 7 3 8 1 6 4 2 5
8 2 1 7 3 9 5 4 6
6 5 9 4 2 8 7 1 3
4 3 7 1 6 5 2 8 9
1
487
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
9 9 9 9 9 9 9 9 9
8 5 5 5 6 6 6 6 6
8 5 3 3 3 3 3 3 6
8 5 3 1 1 1 2 3 6
8 5 3 1 1 1 2 4 6
8 5 2 1 1 1 2 4 6
8 5 2 2 2 2 2 4 7
8 5 4 4 4 4 4 4 7
8 8 7 7 7 7 7 7 7
```

ADAlab Online Judge
Powered by OnlineJudge Version: 20220706-3ff68