4th Endpoint

I am a big fan of parallelograms -- I like all parallelogram shaped objects, including tables, desks, boxes, paper, people, and the Dockland office building in Hamburg, Germany in particular.

Being bad at math, I wonder if you could help me with the following parallelogram problem: given the coordinates of the endpoints of two adjacent sides of a parallelogram, can you tell me the coordinates of the fourth endpoint?

Input Format

A single line of eight integers. The first four integers are the (x, y) coordinates of the two endpoints of the first side. The last four integers are the (x, y) coordinates of the two endpoints of the second side.

Constraints

All coordinates are integers between -1000 and 1000.

Output Format

A single line of the (x, y) coordinates of the fourth endpoint of the parallelogram, separated by a single space.

Sample Input 0

```
0 0 0 1 0 1 1 1
```

Sample Output 0

```
1 0
```

Sample Input 1

```
4 6 1 3 1 3 6 2
```

Sample Output 1

```
9 5
```

Sample Input 2

```
4 6 1 3 6 2 1 3
```

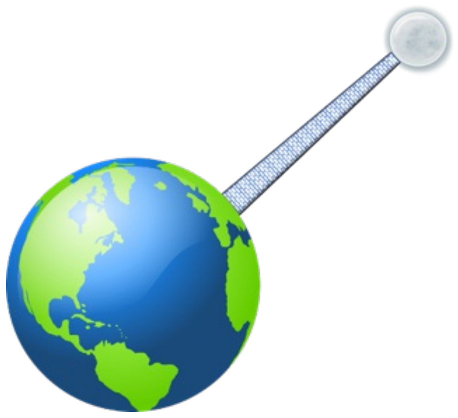
Sample Output 2

```
9 5
```

A Building with a Billion Floors!

Nothing seems impossible to Mr. Paw-see Ball! Ignorance is his strength!

His current project is to build the tallest building on earth. The building will be so tall that it will reach the moon! The moon being at least 225,623 miles away from the earth, the building will have over a billion floors! Think Paw-see Ball?



Mr. Paw-see Ball is a bit superstitious, too (no wonder). He hates 0's as they smell like failures. So, no floor number in the building can have a 0 in it! The floor numbers will go as follows:

1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, ..., 19, 21, ..., 98, 99, 111, 112, ...

You are tasked with writing the software to control the elevators of the building. Given two floor numbers f_s and f_d in that zero-free numbering system, your software must figure out how many floors an elevator must climb to go from floor f_s to floor f_d, that is, compute $|f_d - f_s|$ in a **standard** number system with zeros that the computer can manipulate easily.

Input Format

A single line of two floor numbers (first f_s and then f_d) in the zero-free number system separated by a space.

Constraints

No floor number will have more than 9 digits.

Output Format

Output $f_d - f_s$ in a standard decimal number system with zeros.

Sample Input 0

```
1 11
```

Sample Output 0

```
9
```

Sample Input 1

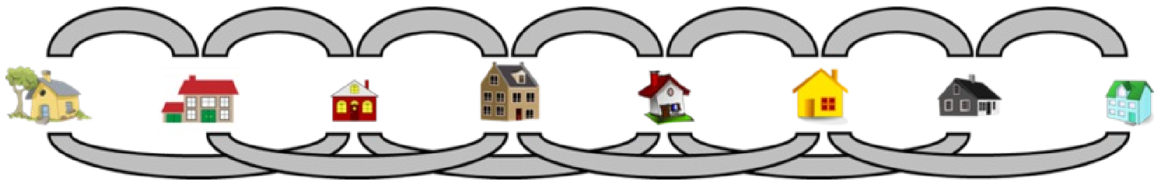
```
111 99
```

Sample Output 1

```
1
```

Roads of Alexandria

Alex and Ria are rebuilding the entire road network of Alexandria. There are N houses in the city which they numbered from 1 to N. Alex's original proposal was to build $N-1$ two-way roads where for $1 \leq i < N$, road i would connect house i with house $i+1$. But Ria pointed out that they would end up building too few roads because if any of those roads became unusable (e.g., due to repairs, flooding, etc.) the city would no longer remain fully connected. So, Alex proposed to build $N-2$ additional two-way roads so that for $1 \leq i < N-1$, a road would directly connect house i with house $i+2$ as well. But this time Ria said that they would end up building more roads than necessary. She drew a graph and showed that even if they did not build some of those $2N-3$ roads each pair of houses would still have an alternate path connecting them when one of the roads became unusable.



Alex and Ria need to choose enough roads to build (from the set of $2N-3$ roads described above) so that the entire city remains connected even if one of those roads is closed. However, given the cost of building each road, they must also make sure that the total cost of building the entire road network is minimized.

Alex and Ria of Alexandria are asking for your help.

Input Format

The first line contains an integer N, denoting the number of houses. The next line contains $N-1$ nonnegative integers, where for $1 \leq i < N$, the i-th integer x_i represents the cost of building a road between house i and house $i+1$. The following line contains $N-2$ nonnegative integers, where for $1 \leq i < N-1$, the i-th integer y_i represents the cost of building a road from house i to house $i+2$.

Constraints

$2 < N \leq 1000$ $0 \leq x_i \leq 100000$ $0 \leq y_i \leq 100000$

Output Format

Output a single integer: the total cost of building the transportation system.

Sample Input 0

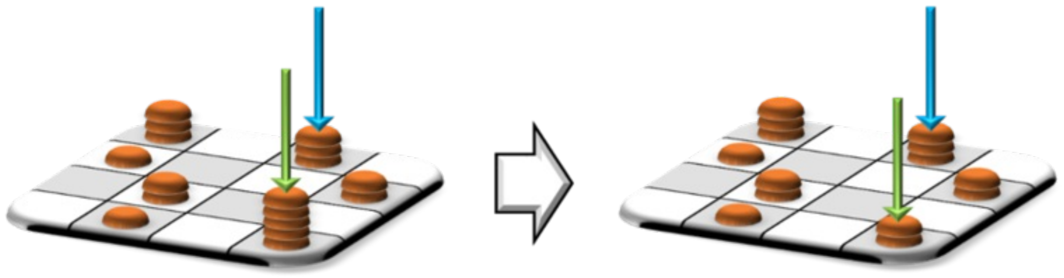
```
2
3
2 3
5
5
2 2 2 2
2 3 2
```

Sample Output 0

```
10
11
```

The Cookie Game

The *cookie game* is a single-player game played on an $n \times n$ grid. Before the game starts cookies are placed in some (or all) grid cells. If multiple cookies are placed in the same cell they are stacked. Then the player starts to eat cookies. Each time the player chooses two grid cells, say, x and y, containing c_x and c_y cookies, respectively, such that $c_x > c_y > 0$. If the player can find such a pair of cells, he/she eats c_y cookies from cell x leaving only $c_x - c_y$ cookies in that cell, but leaves cell y untouched. The player can repeat the process above for as many times as he/she wants. The game ends when the player fails to find such a cell pair.



Given an initial configuration of a game board (i.e., number of cookies in each cell), can you find the maximum number of cookies the player can eat when the game ends?

Input Format

The first line contains the value of n. Each of the next n lines will contain n integers. Line i corresponds to row i of the grid and the j-th integer in that line gives the number of cookies c_{ij} placed in the j-th cell of row i, where $1 \leq i, j \leq n$.

Constraints

- $1 < n \leq 40$
- $0 \leq c_{ij} < 10^6$

Output Format

Output the maximum number of cookies the player can eat on a single line.

Sample Input 0

```
2
1 0
0 2
```

Sample Output 0

```
1
```

Sample Input 1

```
2
6 2
4 0
```

Sample Output 1

```
6
```

Sample Input 2

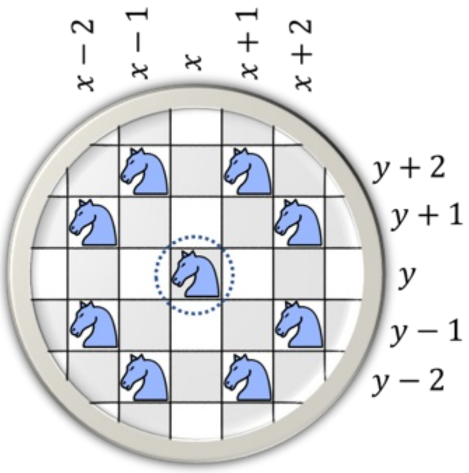
```
3
0 0 18
0 0 0
0 27 0
```

Sample Output 2

```
27
```

Knights on Missions

Two knights, one black and one white, must travel on an $n \times n$ chessboard. Each cell of the chessboard is identified by a unique pair of integers $\langle r, c\rangle$, where r $(1 \le r \le n)$ is the row number and c $(1 \le c \le n)$ is the column number. The black knight starts at cell $\langle r_{bs}, c_{bs}\rangle$ and must go to cell $\langle r_{bt}, c_{bt}\rangle$. The white knight, on the other hand, must go from cell $\langle r_{ws}, c_{ws}\rangle$ to cell $\langle r_{wt}, c_{wt}\rangle$. Both knights start at the same time, and in every step, each makes a move. They must also reach respective destinations at the same time. The goal is to make sure that they complete their journey in minimum number of steps.



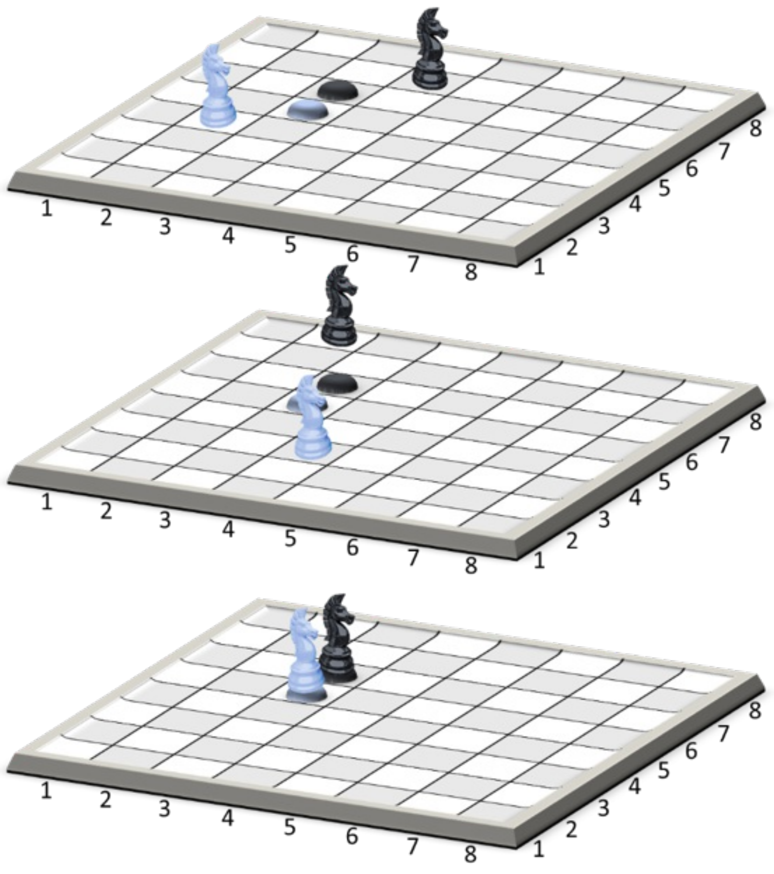
The figure above shows the legal moves of a knight. If the knight is at cell $\langle x, y\rangle$ then it's next move can only be to one of the cells in:

$$M_{x,y} = \{\langle x-1, y+2\rangle, \langle x+1, y+2\rangle, \langle x-2, y+1\rangle, \langle x+2, y+1\rangle,\\ \langle x-2, y-1\rangle, \langle x+2, y-1\rangle, \langle x-1, y-2\rangle, \langle x+1, y-2\rangle\}$$

provided the cell's coordinates are within the boundary of the chessboard. If the black knight is at cell $\langle x, y\rangle$ and the white knight is at one of the cells in $M_{x,y}$ or vice versa, we say that they are in attacking positions.

Both knights are on very important missions and so must avoid fights at all costs. Their starting cells are non-attacking and different, and so are their destination cells. They must also make sure that in every step of their journey they are in non-attacking positions. They must also never occupy the same cell in the same step. Note that if in some intermediate step one of the knights reaches his destination cell but the other does not, then both of them must keep moving until both reach their respective destination cells in the same (final) step.

In the example below, the initial locations of the black and white knights are $\langle 4, 7\rangle$ and $\langle 2, 4\rangle$, respectively, and their target locations are $\langle 3, 6\rangle$ and $\langle 3, 5\rangle$, respectively. They reach their targets in only two steps via cells $\langle 2, 8\rangle$ and $\langle 4, 3\rangle$, respectively. Observe that instead of going via $\langle 2, 8\rangle$ the black knight could have gone through $\langle 5, 5\rangle$. But since $\langle 4, 3\rangle$ and $\langle 5, 5\rangle$ are attacking locations, it avoided that path.



Input Format

The first line contains the size of the board n. The second line contains the integers r_{bs}, c_{bs}, r_{bt}, and c_{bt}. The third line contains the integers r_{ws}, c_{ws}, r_{wt}, and c_{wt}. You may assume that the initial locations of the two knights are different and non-attacking, and so are their final locations.

Constraints

$3 \leq n \leq 64$

Output Format

Output the minimum number of moves required to reach the goal in a single line. But if no solution exists for the test case, just print the line "no solution" (without quotes).

Sample Input 0

```
8
4 1 8 3
4 3 8 1
```

Sample Output 0

```
4
```

Sample Input 1

```
8
4 1 7 3
4 3 8 1
```

Sample Output 1

no solution

Wayboard Shows

Unlike Broadway theater in New York City which has 41 professional theaters to run up to 41 different shows simultaneously, Wayboard theater in New Rock City has only one theater but hundreds of amazing theatrical performances to stage. That's why, Wayboard stages one show per night and usually different shows on different nights. Shows are often repeated (though usually not on consecutive days) but there seems to be no specific pattern in the repetitions. Wayboard publishes its show schedule ahead of time for an entire year or more.

You are such a big fan of Wayboard shows that you often fly to New Rock City to watch them. You stay there for days watching as many different shows as you can. On a visit to New Rock City, if Wayboard stages a show on some night that you already watched during that visit then you consider that night "wasted". During any given visit, you do not want to have more than a fixed number of wasted nights.

Given a Wayboard show schedule and the number k of wasted nights you can tolerate, you want to figure out when you must go to New Rock City so that you can stay there for the maximum number of nights without encountering more than k wasted nights.

Input Format

The input starts with an integer T giving the number of test cases to follow. Each test case starts with a line containing two integers n and k, where n is the number of (consecutive) nights in the given schedule and k is the maximum number of wasted nights you can handle. Each of the next n lines gives the name of the show on a night, where the i-th ($1 \leq i \leq n$) line is for night i assuming that tomorrow night is night 1. Each show name will be between 1 and 20 characters long.

Constraints

- $1 \leq n \leq 100000$
- $0 \leq k \leq 10$

Output Format

For each test case, output two integers on a single line separated by a space. The first integer will give the maximum number of nights you can stay in New Rock City without encountering more than k wasted nights under the given schedule. The second integer will specify on which night you should start watching the shows. If there are multiple solutions, output the one with the earliest start date.

Sample Input 0

```
10 1
Aladdin
The Magicians Wife
Alice in Wonderland
Aladdin
```

```
Shadows
The Lion King
The Magicians Wife
Rainbow Hearts
The Lion King
Purple Pain
```

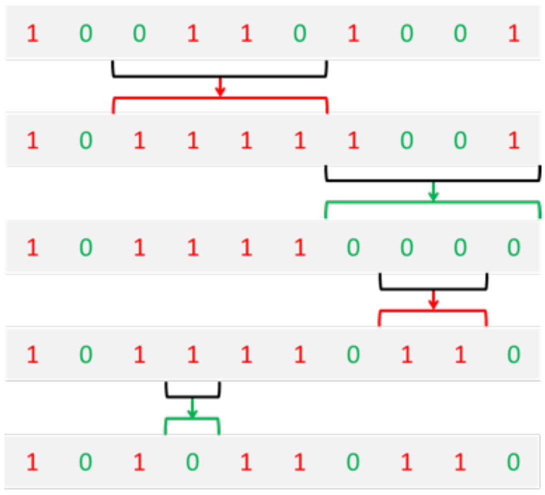
Sample Output 0

```
8 3
```

Robot Gene Therapy

“A long time ago in a galaxy far, far away” an advanced robot civilization arose from the ruins of a biological civilization. Like biological organisms those robots had their own genetic material that controlled their replication, growth and development. Their genomes consisted of DNA strands which were sequences of 0’s and 1’s (unlike, say, our DNA strands which are sequences of four nucleotide bases − adenine, guanine, cytosine, and thymine). Like real biological organisms sometimes they suffered from complicated diseases, but they developed advanced gene therapy approaches to successfully treat many of those.

One of their DNA editing approaches was as follows. They started with a source strand (fragment) and repeatedly applied a very simple operation until they could create the target strand (fragment). The operation involved selecting any one continuous segment of the strand and replacing all symbols in that segment with the same binary digit, that is, either with all 0’s or all 1’s. An example is given below which shows how to transform 1001101001 to 1010110110 in four steps.



Given a pair of source and target robot DNA strand fragments of the same length, how could the robots make the transformation with the fewest number of applications of the operation?

Input Format

The input will start with an integer T giving the number of test cases to follow. Each test case will consist of two lines. The first line will contain the source DNA fragment and the second line will contain the target DNA fragment.

Constraints

- 1 \leq T \leq 200
- The source fragment and the target fragment always have the same length which will not be more than 200.

Output Format

For each test case, output the smallest number of applications of the operation needed for transforming the source to the target on a single line.

Sample Input 0

```
2
00
11
10001
01100
```

Sample Output 0

```
1
2
```