
tasks

| task | izbori | turbo | kule |
|---------------------------------------|--------------------------------------|-----------------------------------|--------------------------------|
| source file | izbori.pas izbori.c izbori.cpp | turbo.pas turbo.c turbo.cpp | kule.pas kule.c kule.cpp |
| input data | stdin | | |
| output data | stdout | | |
| time limit (Intel Celeron 2.66Ghz) | 1 sec | | |
| memory limit (heap) | 32 MB | | |
| memory limit (stack) | 8 MB | | |
| points | 40 | 50 | 60 |
| | 150 | | |

Presidential elections are underway in a small country. There are a total of M citizens choosing between N candidates labeled 1 to N .

Each voter fills his ballot by listing **all the candidates** in order of preference – they put their favorite candidate in the first place, the second one in the second place, ..., the least favored candidate in the last place.

We say that candidate A **defeated** candidate B if the number of voters that put candidate A in front of candidate B is greater than the number of voters that put candidate B in front of candidate A . For a candidate A we define his **score** as the total number of candidates he defeated. We say that a candidate is a **winner of the elections** if his **score** is **greater than or equal** to the score of every other candidate.

Write a program that, given the votes of the citizens, finds the **winner (or winners) of the elections**.

input data

The first line of input contains two integers M and N , $1 \leq M, N \leq 50$, M is **odd**.

Each of the next N lines contains sequence of N numbers, the contents of each citizen's ballot.

output data

If there is only one winner, output his label on the first and only line.

If there are more winners, output their labels **in arbitrary order**, each label in its own line.

examples

input

1 3
1 2 3

output

1

input

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

output

1
4

input

5 5
1 2 3 4 5
5 4 3 2 1
1 2 3 4 5
5 4 3 2 1
3 1 2 4 5

output

3

turbo

A new TV set, besides the regular capability to directly select a channel with the remote control, is equipped with the new "**turbo**" mode with an associated key (labeled with the letter **T**) on the remote control.

Turbo mode makes it easier to cycle through a number of previously selected channels. For usual operation, the user selects a channel by pressing the appropriate key on the remote control.

Let the currently displayed channel be channel X.

After pressing the T key, the TV set enters the turbo mode, and changes **to the first channel displayed after the last change to channel X**. Additional presses of the T key make the TV set cycle through channels that were displayed **between the two appearances of the channel X**, channel X included. In one cycle, the TV set displays each channel only once, in order of their first appearance.

If the first appearance of the channel X is just before entering the turbo mode, the cycle starts with the first selected channel.

The TV set leaves the turbo mode when the user manually selects a channel.

For example, if the pressed keys are 1, 2, 3, T, T, T, T, T, 2, 4, 1, T, T, T i T:

| Step | Pressed key | Explanation | Channel displayed |
|------|-------------|---|-------------------|
| 1. | 1 | Manual select | 1 |
| 2. | 2 | Manual select | 2 |
| 3. | 3 | Manual select | 3 |
| 4. | T | When the TV enters turbo mode, the displayed channel is 3 (first appearance). The turbo cycle starts from the first selected channel (1 in this case) | 1 |
| 5. | T | After channel 1, channel 2 was selected | 2 |
| 6. | T | After channel 2, channel 3 was selected | 3 |
| 7. | T | The cycle started in step 4 is repeated | 1 |
| 8. | T | After channel 1, channel 2 was selected | 2 |
| 9. | 2 | Manual select, the TV set leaves turbo mode | 2 |
| 10. | 4 | Manual select | 4 |
| 11. | 1 | Manual select | 1 |
| 12. | T | The last occurrence of channel 1 was in step 7, the TV set changes to the next displayed channel | 2 |
| 13. | T | If we would display channel 2 from step 9, it will be its second appearance, so we skip it | 4 |
| 14. | T | The channel X is also included in the cycle | 1 |
| 15. | T | The cycle started in step 12 is repeated | 2 |

Write a program that, given the sequence of pressed keys, determines which channel is displayed each time the T key is pressed.

The TV set has 99 channels labeled with numbers from 1 to 99 and the first pressed key will not be T.

turbo

input data

The first line of input contains an integer N , $1 \leq N \leq 10\,000$.

Each of the following N lines contains a number or the letter T.

output data

For each time the T key is pressed, output the channel that is displayed.

examples

input

10
1
2
3
3
4
T
T
T
T
T

output

1
2
3
4
1

input

10
1
2
3
2
3
1
T
T
T
T

output

2
3
1
2

input

15
1
2
3
T
T
T
T
2
4
1
T
T
T
T

output

1
2
3
1
2
2
4
1
2

kule

Two rooks are to be placed on a chess-like $N \times N$ game board. Each square contains a single non-negative integer. The rooks must be placed on **different** squares.

We say that some square on the board is **attacked** if that square is **in the same row** or **in the same column** as one of the rooks. Squares containing the rooks themselves **are not** considered attacked.

We want to place our rooks so that the **total sum** of the numbers on all attacked squares is **as large as possible**.

Write a program that will find this maximum sum.

input data

The first line of input contains an integer N , $2 \leq N \leq 300$.

Each of the following N lines contains N integers. Each number will be greater than or equal to 0 and less than or equal to 1000. These are the numbers on the board.

output data

The first and only line of output should contain a single integer – the maximum sum from the task description.

examples

input

```
3
0 1 4
3 0 2
1 4 1
```

output

```
15
```

input

```
4
0 1 1 1
1 0 4 3
0 1 3 5
0 0 2 5
```

output

```
23
```

input

```
5
4 2 2 3 3
4 2 1 4 0
1 3 4 0 1
4 3 0 2 3
0 0 3 0 4
```

output

```
40
```