UCL Algorithm Contest
Round 1 - 2016

Do not open before the start of the contest.

This page is not really blank.

Problem A
Road Works
2pt

It is winter and as you may have noticed, in winter it rains. A lot. This can lead to floods in some cities where roads have bad water evacuation systems. To overcome this problem next year, the mayor of a very big city with very poorly designed roads wants to make road works on every road. Each road of the city has two lanes making it possible to traverse it in both directions. In order to do the road works, the mayor has decided to block one lane of each road. This makes the road uni-directional but allows to make road works on the blocked lane. Then when the road works are done, they will unblock the lanes that were blocked and block the other ones.

The city is well connected, in the sense that there is a path between every two intersections. The mayor wants to keep the city well connected during the road works. He would like to know if it is possible to do the road works while keeping the city fully connected, that is, so that at any time there is still at least one path from each pair of intersections.

Can you help him?

Input

The first line of the input contains two integers n and m giving the number of intersections and the number of roads, respectively.

Then follow m lines each with two integers x and y representing that intersections x and y are connected by a road.

No road is given more than once in the input and there exists a path between each pair of intersections.

Constraints:

- $1 \leq n \leq 2500$
- $1 \leq m \leq 5000$
- $0 \leq x < y \leq n-1$

Output

A single line with yes if it is possible to orient the roads so that in each of the two road work phases there exists at least one path between each pair of intersections or no otherwise.

Example

Input 1

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

Output 1

```
yes
```

Input 2

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

Output 2

```
no
```

Problem B
PolygonFall Ascension 2pt

If you have ever been to NWERC (North Western European Regional Contest) you know the game TowerFall Ascension. It is a game where each player is an archer and the goal is to kill the others by shooting arrows at them. You want to program a simplistic version of the game where each level consists of a simple square room with some polygonal obstacles. Given the angle at which an arrow is shot and its initial velocity you would like to know if the arrow will colide with an obstable or not.

We assume that the player is currently in the corner of the room $(0,0)$ and shoots the arrow with speed v_0 at angle $0 < \alpha < 90$ in **degrees** (don't forget to convert it to radians in your program). Also, we treat the arrow as a point. If you don't remember your high school physics, the position of the arrow at time t is given by

$$P(t) = \left(v_0 \cdot t \cdot \cos(\alpha), \quad v_0 \cdot t \cdot \sin(\alpha) - \frac{1}{2} \cdot g \cdot t^2\right)$$

For this problem we will use $g = 10$.

Given v_0 and α write an algorithm that determines if the arrow collides with some polygon. In which case you should output the edge of the polygon with which the arrow collides.

Input

The input starts with a line containing a single integer k giving the number of polygonal obstacles in the room. Then follow k groups of lines describing the polygons. The first line in each group gives the number n of vertices of the polygon. Each of the next n lines contains two integers x, y separated by a single space giving the coordinates of the vertices of the polygon. The edges of the poylgon are defined by connecting the first vertex to the second, the second to the third and so on.

The last line of the input contains the initial velocity v_0 and the angle α is degrees.

Constraints: All values are integers such that

- $1 \leq k \leq 10$
- $1 \leq n \leq 10$
- $0 \leq x, y \leq 10$
- $0 < \alpha < 90$

- $0 < v_0 \leq 100$

No two obstacles touch each other and no obstacle touches position $(0, 0)$. The arrow path will never intersect a polygon vertex.

Output

A single line with `no intersection` if there is no intersection or four integers x_1, y_1, x_2, y_2 separated by single spaces giving the endpoints of the edge that is intersected. The order must respect the order of the vertices of the polygon. See the examples for clarity.

Example

Input 1

```
2
4
4 0
6 0
6 2
4 2
4
4 3
6 3
6 5
4 5
10 45
```

Output 1

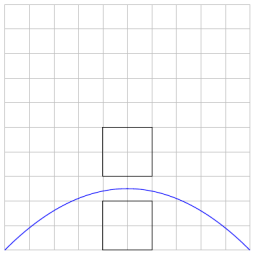
```
no intersection
```

Input 2

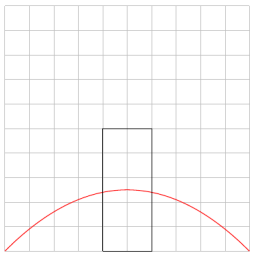
```
1
4
4 0
6 0
6 5
4 5
10 45
```

Output 2

```
4 5 4 0
```



sample input 1



sample input 2

This page is not really blank.

● Problem C
Bomberman
1pt + 1pt

You just acquired the most powerful bombs in Bomberman. If you place a bomb at row i and column j, that bomb will kill all the enemies that lie on that row and on that column. Cool no?

After looking at the positions of the monsters, you realised that if you place a single bomb, the position that kills the most monsters, kills at least two more monster that any other position.

You would like to place k bombs on that $n \times n$ grid so that the maximum number of monsters are killed. Note that you can place a bomb at the same position of a monster.

Input

The first line of the input contains two integers n and k giving the size of the grid and the number of bombs k, respectively.

Then follow n lines each with n integers. The j-th integer of the i has value 1 if there is a monster at position (i, j) and otherwise it has value 0.

Small constraints:

- $1 \leq n \leq 1000$
- $k = 1$

Large constraints:

- $1 \leq n \leq 1000$
- $k = 2$

Output

A single line with the maximum amount of monsters that can be killed.

Example

Input 1

```
5 1
1 1 0 0 0
0 1 0 1 0
1 1 0 0 1
1 0 1 1 1
0 1 1 0 1
```

Output 1

```
8
```

Input 2

```
5 2
1 1 0 0 0
0 1 0 1 0
1 1 0 0 1
1 0 1 1 1
0 1 1 0 1
```

Output 2

```
12
```

Problem D
Collecting gold
1pt + 1pt

You have k robots that you plan to use to collect gold in a goldmine. The goldmine is a $n \times n$ grid and position (i, j) of the grid has $G[i][j]$ gold coins. All your robots start at the left-top position $(0, 0)$ and can only move right (increase i) and down (increase j). Everytime a robot walks into a cell, it collect all the gold in that cell. You are at position $(n-1, n-1)$ where you will collect the gold from the robots.

What is the maximum amount of gold you can collect from the robots?

Input

The input starts with two integers n and k separated by a single space representing the size of the goldmine and the number of robots, respectively.

Then follow n lines giving the amount of gold G in each position. Each line contains n integers and the j-th integer of the i-th line gives $G[i][j]$, the number of gold coins at position (i, j).

Small constraints:

- $1 \leq n \leq 30$
- $k = 1$
- $0 \leq G[i][j] \leq 100$

Large constraints:

- $1 \leq n \leq 30$
- $k = 2$
- $0 \leq G[i][j] \leq 100$

Output

A single line with the maximum amout of gold that can be collected.

Example

Input 1

```
5 1
1 1 1 1 1
1 5 0 0 1
1 0 5 0 1
1 0 0 5 1
1 1 1 1 1
```

Output 1

```
19
```

Input 2

```
5 2
1 1 1 1 1
1 5 0 0 1
1 0 5 0 1
1 0 0 5 1
1 1 1 1 1
```

Output 2

```
26
```

Problem E
Sitting patterns
1pt + 1pt

A set of n chairs are arranged regularly around a circular table. Each chair may be occupied by a woman (W) or a man (M), giving 2^n possible patterns of sexes W and M at the table. If the people all rotate one place around the table, a pattern may change, but after n successive rotations it must recur.

A pattern has period d if rotating the pattern produces exactly d different patterns. Thus a single-sex pattern WW...W or MM...M has period 1, while for even n, the two alternating patterns WMWM...WM and MWMW...MW each have period 2.

How many different patterns of lenght d have period d?

Input

A single line with an integer d as described above.

Small constraints:

$1 \leq d \leq 5$

Large constraints:

$1 \leq d \leq 10^6$

Output

A single line with the total number of different patterns of period d modulo 100000007.

Example

Input 1	Output 1
12	4020

Input 2	Output 2
8	240

This page is not really blank.