1 1984

Task: It is the year 1984 and you work in the Ministry of Peace of Oceania. Oceania is at war against Eastasia. As far as you know Oceania have always been at war against Eastasia. Now, Eastasia is working on a new miracle weapon. To finish the development a train, which is located at train station 1 at the moment has to reach train station B. The railway network consists of n train stations and m rail lines between the train stations. There is at most one rail line between two train stations, and the train can use a rail line in both directions. You have to prevent that the train reaches its target. Thereto, you can destroy rail lines or train stations (apart from station 1 and B). There are costs for the destruction of stations and rails. Compute the minimum cost for preventing the train to reach its target.

Input: The input consists of several test cases, which are concatenated. The description of a single test case begins with a line, which contains two integers. The first number $2 \leq B \leq 50$ denotes the number of train stations, and the second number $0 \leq S \leq 1000$, the number of rail lines. The train stations are numbered from 1 to B. Then, $B - 2$ lines follow: The first integer value denotes the number of the train station and the second integer value the cost of destroying this station. Then S lines follow which describe the rail lines: The first two integer values describe, which train stations are connected by the rail line, and the third integer value denotes the cost of destroying this rail line. You can assume that the cost of a destruction d is bounded by $0 \leq d \leq 10^5$.

The input is ended by a single line, which consists of "0 0".

Output: In the i-th line print the result of the i-th test case.

Sample Input:

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

Sample Output:

2
6

2 Edge orientation

Task: Let $G = (V, E)$ be an undirected graph. For $v \in V$, $\deg^-(v)$ denotes the indegree of v. You are looking for an orientation of the edges, such that $d := \max_{v \in V} \deg^-(v)$ is minimal. Compute d.

Input: There is only one test case specified. The first line contains the number of the nodes n. The nodes are numbered from 1 to n. The second line contains the number of edges m. You can assume that $1 \leq n \leq 500$ and $0 \leq m \leq 2500$. Each of the following m lines contain two integers u_i and v_i, which specify an edge $\{u_i, v_i\}$.

Output: The output consists of a single line, which contains d.

Sample Input 1:

```
2
1
1 2
```

Sample Output 1:

```
1
```

Sample Input 2:

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

Sample Output 2:

```
2
```

3 Plug it in!

Task: Adam just moved into his new apartment and simply placed everything into it at random. This means in particular that he did not put any effort into placing his electronics in a way that each one can have its own electric socket.

Since the cables of his devices have limited reach, not every device can be plugged into every socket without moving it first. As he wants to use as many electronic devices as possible right away without moving stuff around, he now tries to figure out which device to plug into which socket. Luckily the previous owner left behind a plugbar which turns one electric socket into three.

Can you help Adam to figure out how many devices he can power in total?

Input: There will be only a single test case specified in the input. The first line contains the three integegers m, n and k, where m denotes the number of sockets, n the number of electronic devices and k the number of possible connections from devices to sockets. You can assume that $1 \le m \le 1500$, $1 \le n \le 1500$ and $0 \le k \le 75000$. Then, k lines follow each containing two integers x_i and y_i indicating that socket x_i can be used to power device y_i.

Sockets as well as electronic devices are numbered starting from 1.

The plugbar has no cable, i.e. if it is plugged into a socket it simply triples it.

Output: Output one line containing the total number of electrical devices Adam can power.

Sample Input 1:

```
3 6 8
1 1
1 2
1 3
2 3
2 4
3 4
3 5
3 6
```

Sample Output 1:

```
5
```

Sample Input 2:

```
4 5 11
1 1
1 2
1 3
2 1
2 2
2 3
3 1
3 2
3 3
4 4
4 5
```

Sample Output 2:

5

4 Triangula

Task: In the town Triangula new building land should be partitioned into parcels. The building land has the shape of a convex polygon. The parcels should have the shape of triangles, where the vertices of the triangles are vertices of the convex polygon. As usual dispute between the future neighbors grows if the property lines grow. Therefore, you want to compute a partition that minimizes the shared property lines.

Compute the length of the minimum possible shared property line rounded to three decimal places.

Input: There will be only one test case. The first line contains the number n of the vertices $p_1, \ldots, p_n$. The following n lines contain the x- and y-coordinates of the vertices p_i. (The coordinates are given by floating point numbers.) You can assume that $0 \leq x, y \leq 150$. The convex polygon is given by the chain $p_1, \ldots, p_n, p_1$.

Output: The output is the minimum length of the shared property lines rounded to three decimal places.

Sample Input 1:

```
4
0.0 1.0
2.0 0.0
4.0 1.0
2.0 2.0
```

Sample Output 1:

```
2.000
```

Sample Input 2:

```
5
0.0 0.0
2.0 0.0
3.0 2.0
1.0 3.0
0.0 2.0
```

Sample Output 2:

```
5.828
```

5 Is this art or should I clean it up?

Task: You want to honor a great artist. In order to do this, you want to reproduce his pictures as abstract art. You want to convert them into pictures consisting of 16x16 squares of the colors black and white.

You have already written a computer program that gives you the final picture. Now you want to transform a picture consisting of 16x16 black squares into the final picture. For this purpose you can change the color of a square, but then also the colors of the four neighbors on the left, right, above and below invert. Here is an example ("#" denote black, and "O" denote white squares): If you change the color of the square in the center the following will happen:

```
###          #O#
OOO   ->     ###
###          #O#
```

What is the minimum number of changes you have to make to reproduce the final picture?

Input: The input consists of a single test case. The final picture is given by 16 lines containing 16 characters ("#" or "O").

Output: The output is given by the minimum number of changes you have to make to obtain the final picture. If it is impossible to obtain the final picture print -1.

Sample Input:

```
#O##############
OOO#############
#O##############
####OO##########
###O##O#########
####OO##########
################
########O#######
#######OOO######
########O#######
################
################
################
################
################
################
```

Sample Output:

4

6 Edge Orientation 2

Task: You are given an undirected grid graph: Let $m, n \in \mathbb{N}$. The vertices are located at the points $P = \{(i, j) : i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}\}$ and there is an edge between two points $p_1, p_2 \in P$ if and only if $\|p_1 - p_2\|_1 = 1$.

In the following you have to orientate the edges in the grid graph according to the following rule: All edges that lie on the same line must be orientated in the same way. You are given a set $S \subseteq P \times P$. Let $(p_s, p_t) \in S$. We call a path beginning in p_s and ending in p_t *nice* if the path changes its direction at most once. You have to answer the question if there exists an orientation of the edges such that for all pairs in S a nice path exists.

Input: The first line contains the numbers n, m, and $|S|$. Each of the following $|S|$ lines describes one element (p_s, p_t) of S. The first two integers describe the x- and y-coordinate of p_s and the next two integers the x- and y-coordinate of p_t. You can assume that $m, n \leq 20000$ and $|S| \leq 10000$.

Output: If it is possible to orientate the edges in such a way that there is a nice path between every given pair of points, then print "possible". Otherwise, print "impossible".

Sample Input 1:

```
7 7 2
1 1 7 7
7 7 1 1
```

Sample Output 1:

```
possible
```

Sample Input 2:

```
8 8 4
2 2 2 7
7 2 7 7
7 7 2 2
5 4 6 2
```

Sample Output 2:

```
impossible
```

7 Collinearity

Task: You are given n points in $\mathbb{R}^2$. You draw a single line ℓ through $\mathbb{R}^2$. What is the maximum number of points that can lie on ℓ?

Input: There will be only a single test case in a file. The first line denotes the number n of points. In the following n lines the points will be described.

The first number denotes the x-coordinate and the second number denotes the y-coordinate of the point. The coordinates are given as floating point numbers with two decimal places. You can assume that all points are located in $[-100, 100]^2$ and that all points are unique.

Output: Output the maximum number of points that lie on the same line.

Sample Input 1:

```
4
1.50 1.50
2.50 2.50
2.50 1.50
1.50 2.50
```

Sample Output 1:

```
2
```

Sample Input 2:

```
6
-1.00 1.00
-1.00 -1.00
0.00 0.00
0.00 -1.00
-1.00 0.00
-1.00 -3.00
```

Sample Output 2:

```
4
```

8 Separating Points*

Task: This is just a theoretical exercise, so you don't need to code. Nonetheless, we ask you to submit a description of your approach to solve the problem.

You are given n different points in $\mathbb{R}^2$. There is a label from $L = \{1, 2, 3\}$ assigned to each of these points. You want to compute two polygons P_1, P_2, such that P_1 includes all points of a label $a \in L$ and only points of the label a, and P_2 includes all points of a label $b \in L$ with $b \neq a$, and only points of the label b.

How could you compute P_1 and P_2?