A - Penalty Kick

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 100 points

Problem Statement

Takahashi will have N penalty kicks in a soccer match.

For the i-th penalty kick, he will fail if i is a multiple of 3, and succeed otherwise.

Print the results of his penalty kicks.

Constraints

- $1 \leq N \leq 100$
- All inputs are integers.

Input

The input is given from Standard Input in the following format:

```
N
```

Output

Print a string of length N representing the results of Takahashi's penalty kicks. The i-th character $(1 \leq i \leq N)$ should be `o` if Takahashi succeeds in the i-th penalty kick, and `x` if he fails.

Sample Input 1

```
7
```

Sample Output 1

```
ooxooxo
```

Takahashi fails the third and sixth penalty kicks, so the third and sixth characters will be `x`.

Sample Input 2

```
9
```

Sample Output 2

```
ooxooxoox
```

B - Farthest Point

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 200 points

Problem Statement

On the xy-plane, there are N points with ID numbers from 1 to N. Point i is located at coordinates (X_i, Y_i), and no two points have the same coordinates.

From each point, find the farthest point and print its ID number. If multiple points are the farthest, print the smallest of the ID numbers of those points.

Here, we use the Euclidean distance: for two points (x_1, y_1) and (x_2, y_2), the distance between them is $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

Constraints

- $2 \leq N \leq 100$
- $-1000 \leq X_i, Y_i \leq 1000$
- $(X_i, Y_i) \neq (X_j, Y_j)$ if $i \neq j$.
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
X_1 \quad Y_1 \\
X_2 \quad Y_2 \\
\vdots \\
X_N \quad Y_N
\end{array}
$$

Output

Print N lines. The i-th line should contain the ID number of the farthest point from point i.

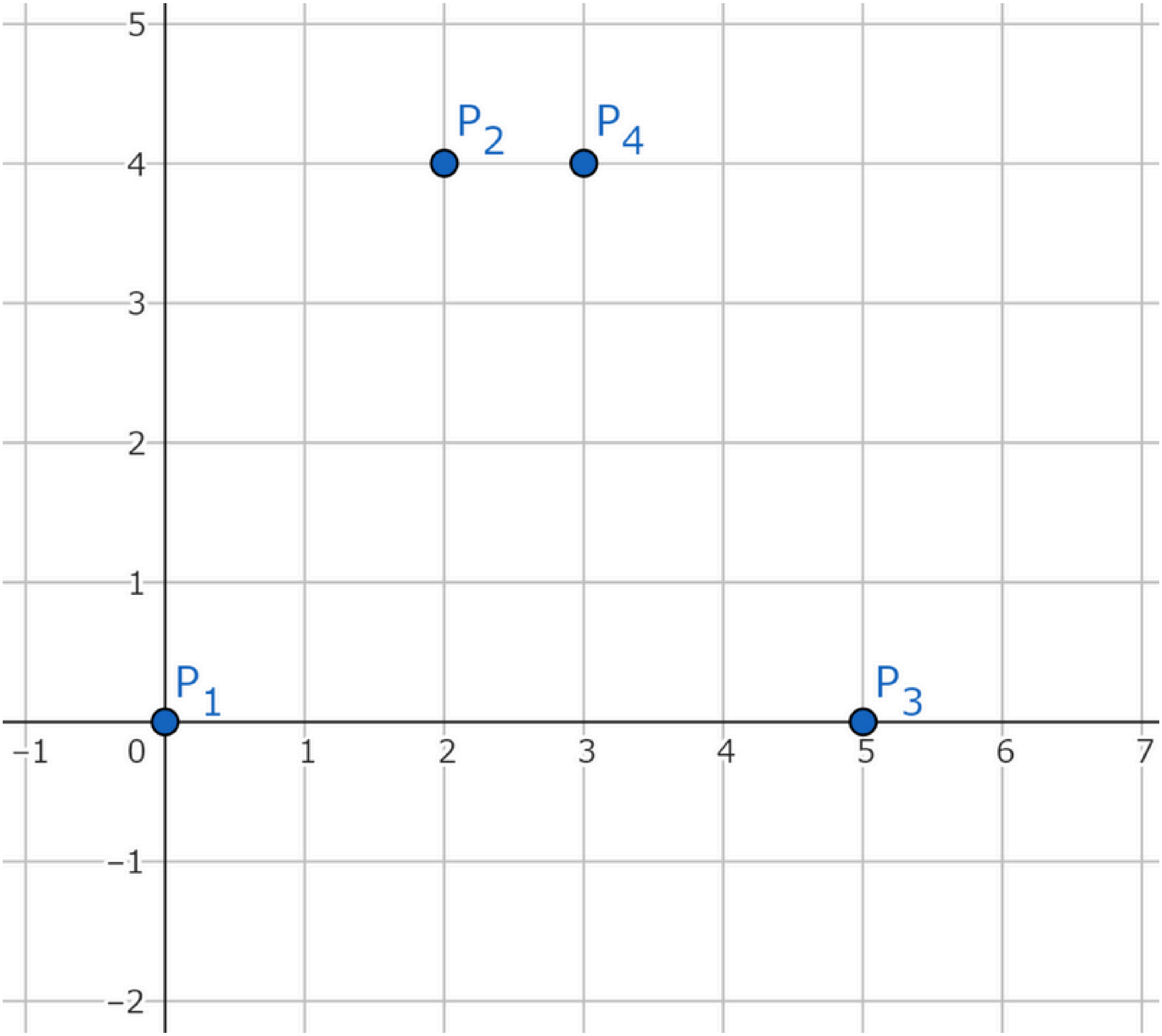
Sample Input 1

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

Sample Output 1

```
3
3
1
1
```

The following figure shows the arrangement of the points. Here, P_i represents point i.



The farthest point from point 1 are points 3 and 4, and point 3 has the smaller ID number.

The farthest point from point 2 is point 3.

The farthest point from point 3 are points 1 and 2, and point 1 has the smaller ID number.

The farthest point from point 4 is point 1.

Sample Input 2

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

Sample Output 2

```
6
6
6
6
6
4
```

C - Colorful Beans

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 250 points

Problem Statement

There are N types of beans, one bean of each type. The i-th type of bean has a deliciousness of A_i and a color of C_i. The beans are mixed and can only be distinguished by color.

You will choose one color of beans and eat one bean of that color. By selecting the optimal color, maximize the minimum possible deliciousness of the bean you eat.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq A_i \leq 10^9$
- $1 \leq C_i \leq 10^9$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
A_1 \quad C_1 \\
A_2 \quad C_2 \\
\vdots \\
A_N \quad C_N
\end{array}
$$

Output

Print as an integer the maximum value of the minimum possible deliciousness of the bean you eat.

Sample Input 1

```
4
100 1
20 5
30 5
40 1
```

Sample Output 1

```
40
```

Note that beans of the same color cannot be distinguished from each other.

You can choose color 1 or color 5.

- There are two types of beans of color 1, with deliciousness of 100 and 40. Thus, the minimum deliciousness when choosing color 1 is 40.
- There are two types of beans of color 5, with deliciousness of 20 and 30. Thus, the minimum deliciousness when choosing color 5 is 20.

To maximize the minimum deliciousness, you should choose color 1, so print the minimum deliciousness in that case: 40.

Sample Input 2

```
10
68 3
17 2
99 2
92 4
82 4
10 3
100 2
78 1
3 1
35 4
```

Sample Output 2

```
35
```

D - Medicines on Grid

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 425 points

Problem Statement

There is a grid with H rows and W columns. Let (i, j) denote the cell at the i-th row from the top and the j-th column from the left. The state of each cell is represented by the character $A_{i,j}$, which means the following:

- `.`: An empty cell.
- `#`: An obstacle.
- `S`: An empty cell and the start point.
- `T`: An empty cell and the goal point.

Takahashi can move from his current cell to a vertically or horizontally adjacent empty cell by consuming 1 energy. He cannot move if his energy is 0, nor can he exit the grid.

There are N medicines in the grid. The i-th medicine is at the empty cell (R_i, C_i) and can be used to **set** the energy **to** E_i. Note that the energy does not necessarily increase. He can use the medicine in his current cell. The used medicine will disappear.

Takahashi starts at the start point with 0 energy and wants to reach the goal point. Determine if this is possible.

Constraints

- $1 \leq H, W \leq 200$
- $A_{i,j}$ is one of `.`, `#`, `S`, and `T`.
- Each of `S` and `T` exists exactly once in $A_{i,j}$.
- $1 \leq N \leq 300$
- $1 \leq R_i \leq H$
- $1 \leq C_i \leq W$
- $(R_i, C_i) \neq (R_j, C_j)$ if $i \neq j$.
- A_{R_i,C_i} is not `#`.
- $1 \leq E_i \leq HW$

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
H \quad W \\
A_{1,1}A_{1,2}\cdots A_{1,W} \\
A_{2,1}A_{2,2}\cdots A_{2,W} \\
\vdots \\
A_{H,1}A_{H,2}\cdots A_{H,W} \\
N \\
R_1 \quad C_1 \quad E_1 \\
R_2 \quad C_2 \quad E_2 \\
\vdots \\
R_N \quad C_N \quad E_N
\end{array}
$$

Output

If Takahashi can reach the goal point from the start point, print `Yes`; otherwise, print `No`.

Sample Input 1

```
4 4
S...
#..#
#...
..#T
4
1 1 3
1 3 5
3 2 1
2 3 1
```

Sample Output 1

```
Yes
```

For example, he can reach the goal point as follows:

- Use medicine 1. Energy becomes 3.
- Move to $(1, 2)$. Energy becomes 2.
- Move to $(1, 3)$. Energy becomes 1.
- Use medicine 2. Energy becomes 5.
- Move to $(2, 3)$. Energy becomes 4.
- Move to $(3, 3)$. Energy becomes 3.
- Move to $(3, 4)$. Energy becomes 2.
- Move to $(4, 4)$. Energy becomes 1.

There is also medicine at $(2, 3)$ along the way, but using it will prevent him from reaching the goal.

Sample Input 2

```
2 2
S.
T.
1
1 2 4
```

Sample Output 2

```
No
```

Takahashi cannot move from the start point.

Sample Input 3

```
4 5
..#..
.S##.
.##T.
.....
3
3 1 5
1 2 3
2 2 1
```

Sample Output 3

```
Yes
```

E - Minimize Sum of Distances

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 475 points

Problem Statement

You are given a tree with N vertices. The vertices are numbered 1 to N, and the i-th edge connects vertices A_i and B_i.

You are also given a sequence of positive integers $C = (C_1, C_2, \ldots, C_N)$ of length N. Let $d(a, b)$ be the number of edges between vertices a and b, and for $x = 1, 2, \ldots, N$, let $f(x) = \sum_{i=1}^{N} (C_i \times d(x, i))$. Find $\min_{1 \leq v \leq N} f(v)$.

Constraints

- $1 \leq N \leq 10^5$
- $1 \leq A_i, B_i \leq N$
- The given graph is a tree.
- $1 \leq C_i \leq 10^9$

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
A_1 \quad B_1 \\
A_2 \quad B_2 \\
\vdots \\
A_{N-1} \quad B_{N-1} \\
C_1 \quad C_2 \quad \cdots \quad C_N
\end{array}
$$

Output

Print the answer in one line.

Sample Input 1

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

Sample Output 1

```
5
```

For example, consider calculating $f(1)$. We have $d(1,1) = 0, d(1,2) = 1, d(1,3) = 1, d(1,4) = 2$. Thus, $f(1) = 0 \times 1 + 1 \times 1 + 1 \times 1 + 2 \times 2 = 6$.

Similarly, $f(2) = 5, f(3) = 9, f(4) = 6$. Since $f(2)$ is the minimum, print `5`.

Sample Input 2

```
2
2 1
1 1000000000
```

Sample Output 2

```
1
```

$f(2) = 1$, which is the minimum.

Sample Input 3

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

Sample Output 3

```
56
```

F - Oddly Similar

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 550 points

Problem Statement

There are N sequences of length M, denoted as $A_1, A_2, \ldots, A_N$. The i-th sequence is represented by M integers $A_{i,1}, A_{i,2}, \ldots, A_{i,M}$.

Two sequences X and Y of length M are said to be similar if and only if the number of indices $i (1 \leq i \leq M)$ such that $X_i = Y_i$ is odd.

Find the number of pairs of integers (i, j) satisfying $1 \leq i < j \leq N$ such that A_i and A_j are similar.

Constraints

- $1 \leq N \leq 2000$
- $1 \leq M \leq 2000$
- $1 \leq A_{i,j} \leq 999$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad M \\
A_{1,1} \quad A_{1,2} \quad \ldots \quad A_{1,M} \\
A_{2,1} \quad A_{2,2} \quad \ldots \quad A_{2,M} \\
\vdots \\
A_{N,1} \quad A_{N,2} \quad \ldots \quad A_{N,M}
\end{array}
$$

Output

Print the answer as an integer.

Sample Input 1

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

Sample Output 1

```
1
```

The pair $(i,j) = (1,2)$ satisfies the condition because there is only one index k such that $A_{1,k} = A_{2,k}$, which is $k = 1$.

The pairs $(i,j) = (1,3), (2,3)$ do not satisfy the condition, making $(1,2)$ the only pair that does.

Sample Input 2

```
6 5
8 27 27 10 24
27 8 2 4 5
15 27 26 17 24
27 27 27 27 27
27 7 22 11 27
19 27 27 27 27
```

Sample Output 2

```
5
```

G - Max (Sum - Max)

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 650 points

Problem Statement

You are given two integer sequences A and B of length N. For $k = 1, 2, \ldots, N$, solve the following problem:

- Consider choosing k distinct integers between 1 and N, inclusive. Let S be the set of chosen integers. Find the maximum value of $(\sum_{i \in S} A_i) - \max_{i \in S} B_i$.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $-10^9 \leq A_i \leq 10^9$
- $-2 \times 10^{14} \leq B_i \leq 2 \times 10^{14}$

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
A_1 \quad B_1 \\
A_2 \quad B_2 \\
\vdots \\
A_N \quad B_N
\end{array}
$$

Output

Print N lines. The i-th line should contain the answer to the problem for $k = i$.

Sample Input 1

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

Sample Output 1

```
3
5
6
```

The following choices are optimal.

- $k = 1$: $S = \{1\}$
- $k = 2$: $S = \{1, 3\}$
- $k = 3$: $S = \{1, 2, 3\}$

Sample Input 2

```
2
0 1
0 1
```

Sample Output 2

```
-1
-1
```

Sample Input 3

```
6
9 7
2 4
7 1
-1000 0
3 4
8 5
```

Sample Output 3

```
6
10
17
20
22
-978
```