A - Strictly Increasing?

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

You are given a positive integer N and a sequence of positive integers $A = (A_1, A_2, \ldots, A_N)$ of length N.

Determine whether A is strictly increasing, that is, whether $A_i < A_{i+1}$ holds for every integer i with $1 \leq i < N$.

Constraints

- $2 \leq N \leq 100$
- $1 \leq A_i \leq 1000$ $(1 \leq i \leq N)$
- All input values are integers.

Input

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

N
$A_1 \quad A_2 \quad \ldots \quad A_N$

Output

If A is strictly increasing, print `Yes`; otherwise, print `No`.

The judge is case-insensitive. For example, if the correct answer is `Yes`, any of `yes`, `YES`, and `yEs` will be accepted.

Sample Input 1 Copy

```
3
1 2 5
```

Sample Output 1 Copy

```
Yes
```

$A_1 < A_2$ and $A_2 < A_3$, so A is strictly increasing.

Sample Input 2 Copy

```
3
3 9 5
```

Sample Output 2 Copy

```
No
```

$A_1 < A_2$, but $A_2 < A_3$ does not hold, so A is not strictly increasing.

Sample Input 3 Copy

```
10
1 1 2 3 5 8 13 21 34 55
```

Sample Output 3 Copy

```
No
```

$A_1 < A_2$ does not hold, so A is not strictly increasing.

B - Make Target

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 200 points

Problem Statement

Overview: Create an $N \times N$ pattern as follows.

```
###########
#.........#
#.#######.#
#.#.....#.#
#.#.###.#.#
#.#.#.#.#.#
#.#.###.#.#
#.#.....#.#
#.#######.#
#.........#
###########
```

You are given a positive integer N.

Consider an $N \times N$ grid. Let (i, j) denote the cell at the i-th row from the top and the j-th column from the left. Initially, no cell is colored.

Then, for $i = 1, 2, \ldots, N$ in order, perform the following operation:

- Let $j = N + 1 - i$.
- If $i \leq j$, fill the rectangular region whose top-left cell is (i, i) and bottom-right cell is (j, j) with black if i is odd, or white if i is even. If some cells are already colored, overwrite their colors.
- If $i > j$, do nothing.

After all these operations, it can be proved that there are no uncolored cells. Determine the final color of each cell.

Constraints

- $1 \leq N \leq 50$
- All input values are integers.

Input

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

N

Output

Print N lines. The i-th line should contain a length-N string S_i representing the colors of the i-th row of the grid after all operations, as follows:

- If cell (i, j) is finally colored black, the j-th character of S_i should be `#`.
- If cell (i, j) is finally colored white, the j-th character of S_i should be `.`.

Sample Input 1 Copy

```
11
```

Sample Output 1 Copy

```
###########
#.........#
#.#######.#
#.#.....#.#
#.#.###.#.#
#.#.#.#.#.#
#.#.###.#.#
#.#.....#.#
#.#######.#
#.........#
###########
```

This matches the pattern shown in the **Overview**.

Sample Input 2 Copy

```
5
```

Sample Output 2 Copy

```
#####
#...#
#.#.#
#...#
#####
```

Colors are applied as follows, where `?` denotes a cell not yet colored:

```
        i=1      i=2      i=3      i=4      i=5
?????   #####    #####    #####    #####    #####
?????   #####    #...#    #...#    #...#    #...#
????? -> ##### -> #...# -> #.#.# -> #.#.# -> #.#.#
?????   #####    #...#    #...#    #...#    #...#
?????   #####    #####    #####    #####    #####
```

Sample Input 3 Copy

```
8
```

Sample Output 3 Copy

```
########
#......#
#.####.#
#.#..#.#
#.#..#.#
#.####.#
#......#
########
```

Sample Input 4 Copy

```
2
```

Sample Output 4 Copy

```
##
##
```

C - Shortest Duplicate Subarray

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 300 points

Problem Statement

You are given a positive integer N and an integer sequence $A = (A_1, A_2, \ldots, A_N)$ of length N.

Determine whether there exists a non-empty (contiguous) subarray of A that has a repeated value, occurring multiple times in A. If such a subarray exists, find the length of the shortest such subarray.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq A_i \leq 10^6$ $(1 \leq i \leq N)$
- All input values are integers.

Input

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

N
$A_1 \quad A_2 \quad \ldots \quad A_N$

Output

If there is no (contiguous) subarray satisfying the condition in the problem statement, print `-1` . Otherwise, print the length of the shortest such subarray.

Sample Input 1 Copy

```
5
3 9 5 3 1
```

Sample Output 1 Copy

```
4
```

$(3, 9, 5, 3)$ and $(3, 9, 5, 3, 1)$ satisfy the condition. The shorter one is $(3, 9, 5, 3)$, which has length 4.

Sample Input 2 Copy

```
4
2 5 3 1
```

Sample Output 2 Copy

```
-1
```

There is no subarray that satisfies the condition.

Sample Input 3 Copy

```
10
1 1 2 3 5 8 13 21 34 55
```

Sample Output 3 Copy

```
2
```

D - Pigeon Swap

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 350 points

Problem Statement

There are N pigeons labeled $1, 2, \ldots, N$ and N nests labeled $1, 2, \ldots, N$.

Initially, pigeon i $(1 \leq i \leq N)$ is in nest i.

You will perform Q operations on these pigeons.

There are three types of operations:

- Type 1: You are given integers a and b $(1 \leq a \leq N, 1 \leq b \leq N)$. Take pigeon a out of its current nest and move it to nest b.
- Type 2: You are given integers a and b $(1 \leq a < b \leq N)$. Move all pigeons in nest a to nest b, and move all pigeons in nest b to nest a. These two moves happen simultaneously.
- Type 3: You are given an integer a $(1 \leq a \leq N)$. Pigeon a reports the label of the nest it is currently in.

Print the result of each Type 3 operation in the order the operations are given.

Constraints

- $1 \leq N \leq 10^6$
- $1 \leq Q \leq 3 \times 10^5$
- Every operation is of Type $1, 2$, or 3.
- All operations are valid according to the problem statement.
- There is at least one Type 3 operation.
- All input values are integers.

Input

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

$$
\begin{aligned}
&N \quad Q \\
&op_1 \\
&op_2 \\
&\vdots \\
&op_Q
\end{aligned}
$$

Here, op_i on the line $i+1$ represents the i-th operation in one of the following formats.

If the i-th operation is of Type 1, op_i is in the following format:

$$1 \quad a \quad b$$

This corresponds to a Type 1 operation with the given integers being a and b.

If the i-th operation is of Type 2, op_i is in the following format:

$$2 \quad a \quad b$$

This corresponds to a Type 2 operation with the given integers being a and b.

If the i-th operation is of Type 3, op_i is in the following format:

$$3 \quad a$$

This corresponds to a Type 3 operation with the given integer being a.

Output

Let the number of Type 3 operations be q. Print q lines. The i-th line $(1 \leq i \leq q)$ should contain the nest number reported in the i-th Type 3 operation.

Sample Input 1 Copy

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

Sample Output 1

Copy

```
4
5
2
5
```

In the operations given, the pigeons move as shown in the figure below:

The nest numbers to be reported for the Type 3 operations are $4, 5, 2, 5$. Print `4`, `5`, `2`, `5` on separate lines.

Sample Input 2 Copy

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

Sample Output 2 Copy

```
1
```

The destination nest of a Type 1 operation may be the same as the nest the pigeon is currently in.

Sample Input 3 Copy

```
30 15
3 3
2 8 30
2 12 15
2 2 17
1 19 1
2 7 30
3 12
3 8
2 25 26
1 13 10
1 16 10
2 16 29
2 1 21
2 6 11
1 21 8
```

Sample Output 3 Copy

```
3
15
7
```

E - Flip Edge

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 425 points

Problem Statement

You are given a directed graph with N vertices and M edges. The i-th edge $(1 \leq i \leq M)$ is a directed edge from vertex u_i to vertex v_i.

Initially, you are at vertex 1. You want to repeat the following operations until you reach vertex N:

- Perform one of the two operations below:
  - Move along a directed edge from your current vertex. This incurs a cost of 1. More precisely, if you are at vertex v, choose a vertex u such that there is a directed edge from v to u, and move to vertex u.
  - Reverse the direction of all edges. This incurs a cost of X. More precisely, if and only if there was a directed edge from v to u immediately before this operation, there is a directed edge from u to v immediately after this operation.

It is guaranteed that, for the given graph, you can reach vertex N from vertex 1 by repeating these operations.

Find the minimum total cost required to reach vertex N.

Constraints

- $2 \leq N \leq 2 \times 10^5$
- $1 \leq M \leq 2 \times 10^5$
- $1 \leq X \leq 10^9$
- $1 \leq u_i \leq N$ $(1 \leq i \leq M)$
- $1 \leq v_i \leq N$ $(1 \leq i \leq M)$
- For the given graph, it is guaranteed that you can reach vertex N from vertex 1 by the operations described.
- All input values are integers.

Input

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

$$
\begin{array}{lll}
N & M & X \\
u_1 & v_1 & \\
u_2 & v_2 & \\
\vdots & & \\
u_M & v_M &
\end{array}
$$

Output

Print the minimum total cost required to reach vertex N.

Sample Input 1

Copy

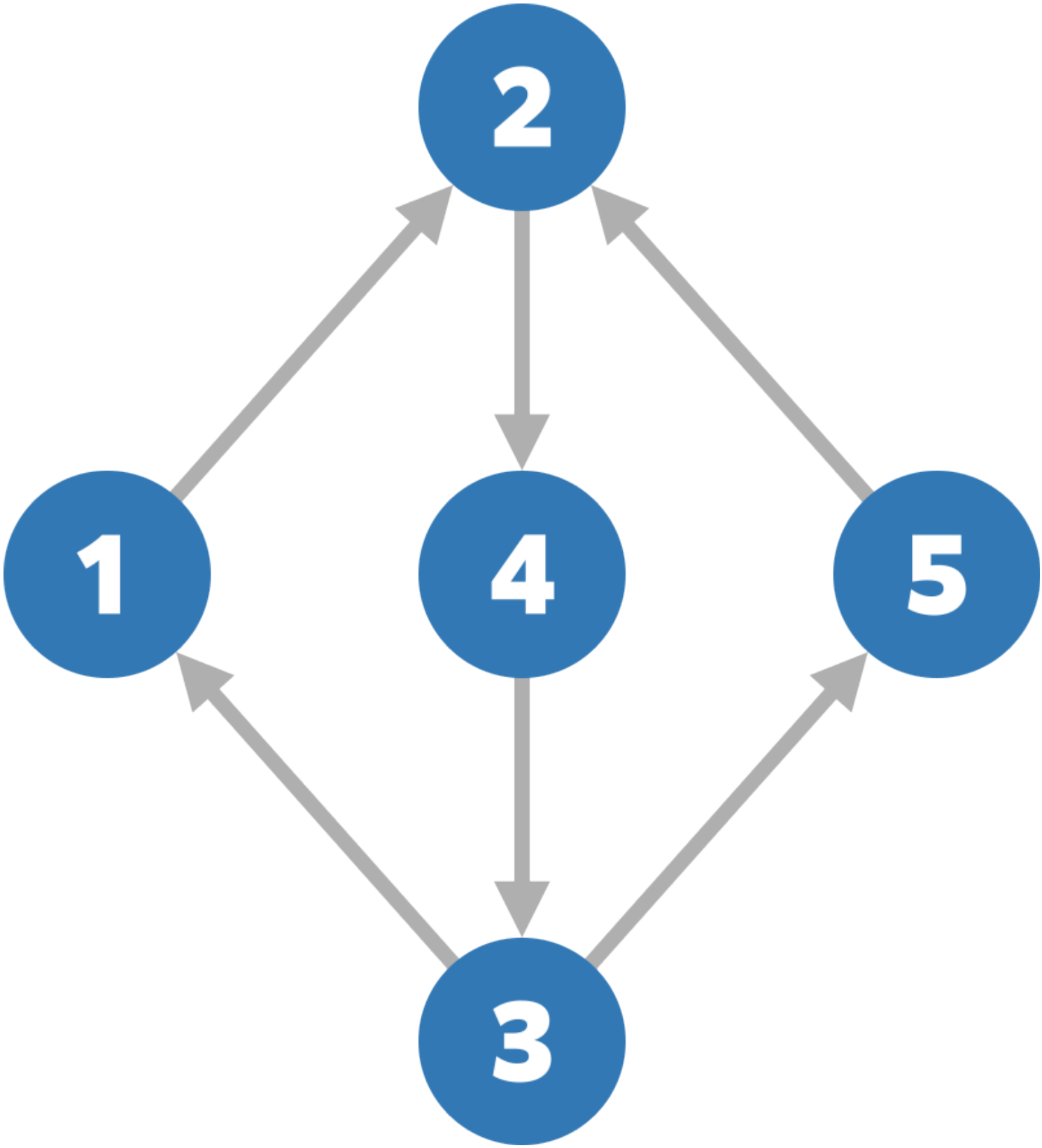
```
5 6 5
1 2
2 4
3 1
3 5
4 3
5 2
```

Sample Output 1

Copy

```
4
```

The given graph looks like this:



You can reach vertex 5 with a total cost of 4 by doing the following:

- Move to vertex 2 at a cost of 1.
- Move to vertex 4 at a cost of 1.
- Move to vertex 3 at a cost of 1.
- Move to vertex 5 at a cost of 1.

It is impossible to reach vertex 5 with a total cost of 3 or less, so print `4`.

Sample Input 2 Copy

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

Sample Output 2 Copy

```
3
```

The graph is the same as in Sample 1, but the cost to reverse edges is different.

You can reach vertex 5 with a total cost of 3 as follows:

- Move to vertex 2 at a cost of 1.
- Reverse all edges at a cost of 1.
- Move to vertex 5 at a cost of 1.

It is impossible to reach vertex 5 with a total cost of 2 or less, so print `3`.

Sample Input 3 Copy

```
8 7 613566756
2 1
2 3
4 3
4 5
6 5
6 7
8 7
```

Sample Output 3 Copy

```
4294967299
```

Note that the answer may exceed the 32-bit integer range.

Sample Input 4 Copy

```
20 13 5
1 3
14 18
18 17
12 19
3 5
4 6
13 9
8 5
14 2
20 18
8 14
4 9
14 8
```

Sample Output 4 Copy

```
21
```

F - Smooth Occlusion

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 500 points

Problem Statement

Takahashi has $2N$ teeth: N upper teeth and N lower teeth.

The length of the i-th upper tooth from the left ($1 \leq i \leq N$) is U_i, and the length of the i-th lower tooth from the left ($1 \leq i \leq N$) is D_i.

His teeth are said to "fit together well" if both of the following conditions are satisfied:

- There exists an integer H such that $U_i + D_i = H$ for every integer i with $1 \leq i \leq N$.
- $|U_i - U_{i+1}| \leq X$ for every integer i with $1 \leq i < N$.

He can perform the following operation any number of times:

- Pay 1 yen to use a tooth-grinding machine, choose exactly one tooth whose length is positive, and reduce its length by 1.

No other method may be used to change the lengths of the teeth.

Find the minimum total amount of money he needs to pay to make his teeth fit together well.

Constraints

- $2 \leq N \leq 2 \times 10^5$
- $1 \leq U_i \leq 10^9$ $(1 \leq i \leq N)$
- $1 \leq D_i \leq 10^9$ $(1 \leq i \leq N)$
- $1 \leq X \leq 10^9$
- All input values are integers.

Input

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

$$
\begin{array}{ll}
N & X \\
U_1 & D_1 \\
U_2 & D_2 \\
\vdots & \\
U_N & D_N
\end{array}
$$

Output

Print the answer.

Sample Input 1 Copy

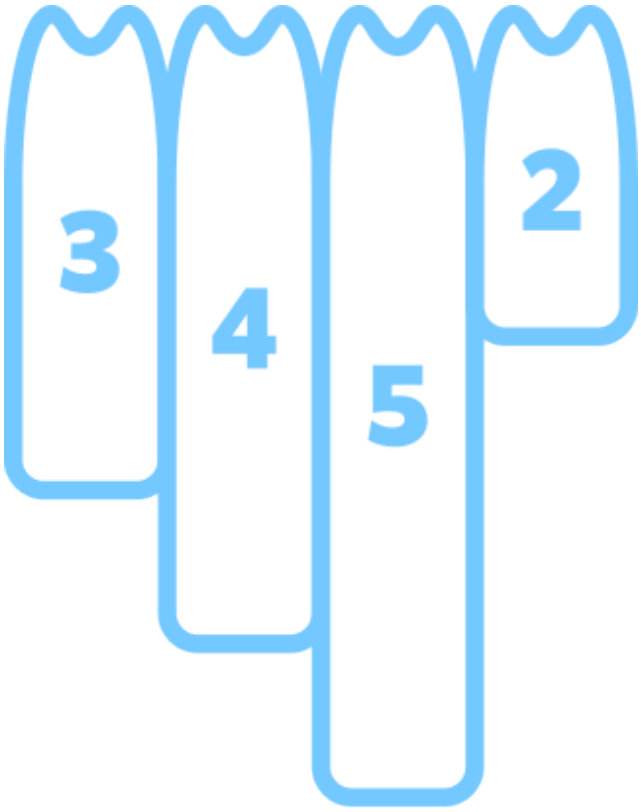
```
4 3
3 1
4 1
5 9
2 6
```

Sample Output 1

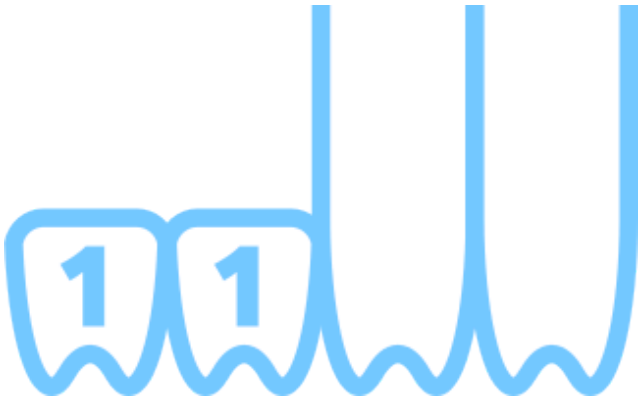
Copy

```
15
```

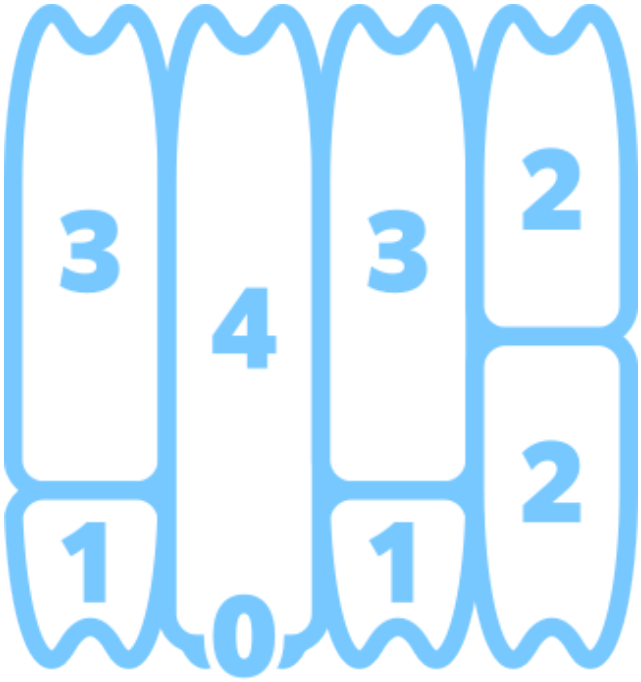
Initially, Takahashi's teeth have the following lengths:


3
4
5
2


9
6



For example, you can make them fit together well in the following way:



It costs 15 yen to achieve these lengths.

It is impossible to make them fit together well with 14 yen or less, so print `15`.

Sample Input 2 Copy

```
4 1000000000
3 3
3 3
3 3
3 3
```

Sample Output 2 Copy

```
0
```

It is possible that the teeth already fit together well without any changes.

Sample Input 3 Copy

```
4 1
1000000000 1000000000
1000000000 1000000000
1000000000 1000000000
1 1
```

Sample Output 3 Copy

```
5999999994
```

Note that the answer may exceed the 32-bit integer range.

Sample Input 4 Copy

```
15 128
748 169
586 329
972 529
432 519
408 587
138 249
656 114
632 299
984 755
404 772
155 506
832 854
353 465
387 374
567 385
```

Sample Output 4 Copy

```
9460
```

G - Minimum Steiner Tree 2

Time Limit: 4 sec / Memory Limit: 1024 MB

Score : 625 points

Problem Statement

You are given positive integers N, K, and an $N \times N$ matrix $C = (C_{i,j})\ (1 \leq i, j \leq N)$. It is guaranteed that $C_{i,i} = 0$ and $C_{i,j} = C_{j,i}$.

There is a weighted complete graph with N vertices labeled $1, 2, \ldots, N$. For each pair of vertices i and j with $i < j$, the weight of the edge connecting vertices i and j is $C_{i,j}$.

You have Q queries. In the i-th query, you are given a pair of distinct integers s_i and t_i (both between $K + 1$ and N, inclusive), for which you must solve the following problem.

> A **good graph** is a **connected** graph containing all of the vertices $1, 2, \ldots, K, s_i, t_i$ obtained by removing any number of edges and vertices (possibly zero) from the graph above.
>
> The **cost** of a good graph is the sum of the weights of its edges.
>
> Find the minimum possible cost of a good graph.

Constraints

- $3 \leq N \leq 80$
- $1 \leq K \leq \min(N - 2, 8)$
- $0 \leq C_{i,j} \leq 10^9\ (1 \leq i, j \leq N, i \neq j)$
- $C_{i,j} = C_{j,i}\ (1 \leq i, j \leq N, i \neq j)$
- $C_{i,i} = 0\ (1 \leq i \leq N)$
- $1 \leq Q \leq 5000$
- $K + 1 \leq s_i, t_i \leq N\ (1 \leq i \leq Q)$
- $s_i \neq t_i\ (1 \leq i \leq Q)$
- All input values are integers.

Input

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

$$
\begin{aligned}
&N \quad K \\
&C_{1,1} \quad C_{1,2} \quad \dots \quad C_{1,N} \\
&C_{2,1} \quad C_{2,2} \quad \dots \quad C_{2,N} \\
&\vdots \\
&C_{N,1} \quad C_{N,2} \quad \dots \quad C_{N,N} \\
&Q \\
&s_1 \quad t_1 \\
&s_2 \quad t_2 \\
&\vdots \\
&s_Q \quad t_Q
\end{aligned}
$$

Output

Print Q lines.

The i-th line should contain the answer for the i-th query.

Sample Input 1 Copy

```
5 2
0 395 395 1 1
395 0 1 395 1
395 1 0 395 1
1 395 395 0 1
1 1 1 1 0
3
3 4
3 5
4 5
```

Sample Output 1

Copy

```
4
3
3
```

Let "edge $i - j$" denote the edge connecting vertices i and j.

For the first query, it is optimal to keep the following vertices and edges, and remove all others. The cost is 4.

- Vertices $1, 2, 3, 4, 5$
- Edges $1 - 4, 2 - 3, 3 - 5, 4 - 5$

For the second query, it is optimal to keep the following vertices and edges, and remove all others. The cost is 3.

- Vertices $1, 2, 3, 5$
- Edges $1 - 5, 2 - 3, 3 - 5$

Sample Input 2

Copy

```
9 5
0 344670307 744280520 967824729 322288793 152036485 628902494 596982638 853214705
344670307 0 249168130 769431650 532405020 981520310 755031424 86416231 284114341
744280520 249168130 0 80707350 256358888 620411718 713892371 272961036 836365490
967824729 769431650 80707350 0 45539861 298766521 722757772 623807668 366719378
322288793 532405020 256358888 45539861 0 361324668 69837030 222135106 935147464
152036485 981520310 620411718 298766521 361324668 0 486834509 225447688 859904884
628902494 755031424 713892371 722757772 69837030 486834509 0 434140395 490910900
596982638 86416231 272961036 623807668 222135106 225447688 434140395 0 22078599
853214705 284114341 836365490 366719378 935147464 859904884 490910900 22078599 0
20
9 7
8 9
9 8
7 6
9 8
8 9
9 7
7 8
7 8
6 9
9 8
9 6
8 6
8 9
6 8
8 7
7 6
8 9
7 6
9 6
```

Sample Output 2

Copy

```
849002970
779165940
779165940
882119751
779165940
779165940
849002970
826924371
826924371
834361320
779165940
834361320
812282721
779165940
812282721
826924371
882119751
779165940
882119751
834361320
```