Azugand City (azugand)

Azugand, the city from the famous Prahovand Valley, is known for its peculiar structure: it has N intersections, numbered from 1 to N, all of which has an assigned value V_i. Two distinct intersections have a street between them if and only if the bitwise AND (represented as &) of their values is non-zero.

Formally, you are given a graph with N vertices, each vertex having a value V_i assigned to it. We have an edge between two distinct vertices i and j iff $V_i \ \& \ V_j \neq 0$.

Figure 1: Welcome to Azugand city!

Andrei, a well-known child from the valley, wants to make a table with the shortest distances between two points of interest, but he is overwhelmed by the number of queries he wants to solve, so he asks for your help! You are given Q queries, in which you have to compute $cost(X, Y)$. We define $cost(X, Y)$ as the minimum number of **edges** that are in the shortest path in the graph between vertices X and Y. If we can't reach vertex Y from vertex X, then the value of $cost(X, Y)$ is -1.

Input

The first line of input contains two integers N and Q, the number of vertices in the graph, and the number of queries, respectively.

The next line contains N integers $V_1, V_2, \ldots, V_N$, the values assigned to the vertices of the graph.

Each of the next Q lines of the input contains two distinct integers X_i and Y_i, the vertices for which you must compute $cost(X_i, Y_i)$.

Output

Print Q integers, each one on a separate line: the value of $cost(X_i, Y_i)$ for each query.

Constraints

- $1 \leq N \leq 200\,000$.
- $1 \leq Q \leq 200\,000$.
- $0 \leq V_i < 2^{20}$ for each $i = 1 \ldots N$.
- $1 \leq X_i \neq Y_i \leq N$ for each $i = 1 \ldots Q$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (7 points) $N \leq 500, Q \leq 500$.

– **Subtask 3** (23 points) $Q \leq 1$.

– **Subtask 4** (21 points) $V_i < 2^5$ for each $i = 1 \ldots N$.

– **Subtask 5** (49 points) No additional limitations.

Examples

input	output

`4 4`
`9 3 16 6`
`1 2`
`2 4`
`4 1`
`2 3`
	`1`
	`1`
	`2`
	`-1`

`7 5`
`3072 5120 67584 73728 49152 24576 40960`
`2 5`
`7 3`
`1 6`
`5 6`
`7 2`
	`5`
	`2`
	`3`
	`1`
	`4`

Explanation

In the **first sample case**:

- In the first query $V_1 = 9$ and $V_2 = 3$, $9\&3 = 1$, so there is an edge between vertices 1 and 2, so the minimum distance is 1.
- In the second query $V_2 = 3$ and $V_4 = 6$, $3\&6 = 2$, so there is an edge between vertices 2 and 4, so the minimum distance is 1.
- In the third query $V_4 = 6$ and $V_1 = 9$, $6\&9 = 0$, so there is no edge between vertices 4 and 1, so the minimum distance is at least 2. For the path $4, 2, 1$ the values are $6, 3, 9$ respectively which means that there is an edge between vertices 4 and 2 also between 2 and 1, so there is a path of length 2 between vertices 4 and 1.
- In the fourth query $V_1\&V_3 = 0$, $V_2\&V_3 = 0$ and $V_4\&V_3 = 0$, which means that there are no edges connected to vertex 3, so there is no path between vertices 2 and 3.