Pizza Orders (`pizza`)

A restaurant has N available ingredients for making pizzas. The ingredients are numbered from 0 to $N - 1$. The menu features M different pizzas, each with a specific list of ingredients.
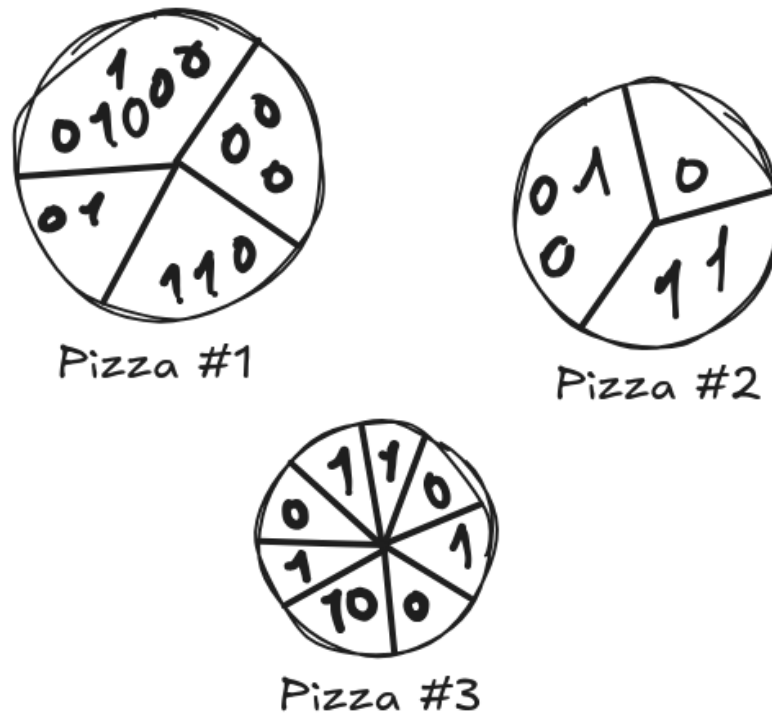

Figure 1: Handcrafted pizzas.

For any given pizza, the restaurant can:

- Add ingredient i for a cost of A_i coins.

- Remove ingredient i for a cost of B_i coins.

You are given Q queries. In each query, determine the minimum cost required to modify any existing pizza into a specific target pizza.

> ☞ Among the attachments of this task you may find a template file `pizza.*` with a sample incomplete implementation.

Input

The first line of the input consists of three space-separated integers N, M, and Q, denoting the number of ingredients, the number of pizzas, and the number of queries, respectively.

The next N lines describe the cost of modifying each ingredient. Each line consists of two space-separated integers A_i and B_i ($0 \leq i \leq N - 1$).

The following $2 \cdot M$ lines describe the pizzas on the menu. Each pizza is represented by two consecutive lines in the following form:

- K – the number of ingredients.

- P_0, P_1, ..., P_{K-1} – the numbers representing each ingredient.

The following $2 \cdot Q$ lines contain the description of the queries, given in the same format as the description of the pizzas.

Output

Output Q lines, the j-th of which should contain a single integer denoting the answer to the j-th query.

Constraints

- $1 \leq N \leq 20$.
- $1 \leq M, Q \leq \min(200\,000,\ 2^N)$.
- $0 \leq A_i, B_i \leq 10^9$ for each $i = 0 \ldots N - 1$.
- $1 \leq K \leq N$.
- $0 \leq P_i \leq N - 1$ for each $i = 0 \ldots K - 1$.
- $P_{i-1} < P_i$ for each $i = 1 \ldots K - 1$.
- All the M pizzas are different.

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** (15 points) $M, Q \leq \min(1000,\ 2^N)$.

– **Subtask 3** (25 points) $N \leq 16$.

– **Subtask 4** (60 points) No additional limitations.

Examples

input	output
3 2 3 5 2 3 3 0 10 2 0 1 1 2 1 0 2 0 1 3 0 1 2	3 0 0

Explanation

In the **sample case**, there are 3 types of ingredients. The cost of adding each of them is 5, 3, and 0 coins, and the cost of removing each of them is 2, 3, and 10 coins, respectively.

There are 2 different pizzas on the menu. The first one has two ingredients: ingredient 0 and ingredient 1; and the second pizza has only one ingredient: ingredient 2.

You are asked to create 3 pizzas:

- In the first query, you have to pay 3 coins to transform one of the two pizzas at the restaurant into a pizza with only ingredient 0.

- In the second query, the specified pizza already exists, so you don't have to pay anything.

- In the third query, you can pay 0 coins and add the third ingredient to the first pizza, obtaining the requested pizza.