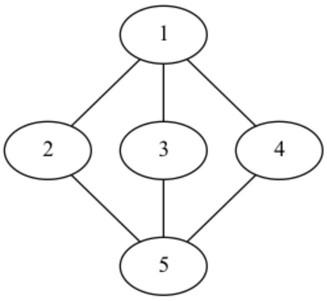
Problem Hide and seek

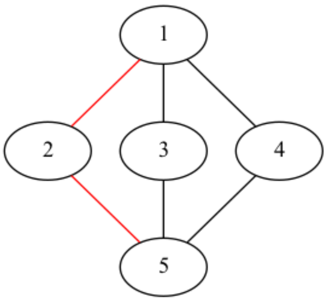
C++ header `hideandseek.h`

Tanaka and Lulu are a bit nervous for the cook-off contest (after all, it is the most important contest in all of Info(1)cup Kingdom). So, in order to relax, they decided to play a game of hide-and-seek. Because they are as competitive as they are, they are playing this game on the largest surface available: the entire Info(1)cup Kingdom. The Info(1)cup Kingdom can be viewed as a network of n towns and m two-way roads. Two towns may be connected by *at most one road*, and *no road connects a town to itself*. It is possible to reach any town from any other town. Lulu and Tanaka will hide in towns.

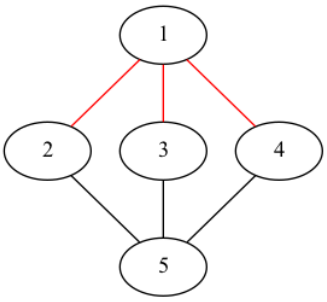
Lulu and Tanaka have k friends which will try to stop them from finding each other. The kingdom has a special property: as long as Lulu and Tanaka hide in different towns, the k friends can each block a road, such that there is no way for Lulu to reach Tanaka. Friends are allowed to do nothing (to not block a road). For example, suppose that the kingdom has 5 towns, with roads between towns $(1,2),(1,3),(1,4),(2,5),(3,5),(4,5)$. The kingdom is pictured below:



In this case it would be possible for them to have $k=3$ friends. If Lulu or Tanaka hide in towns $2,3$ or 4, then their friends could block the 2 roads that are connected to these towns. For example, if one of them hides in 2, by blocking the roads marked with red below, their friends can make it so that Lulu cannot reach Tanaka:



On the other hand, if Lulu hides in town 1, and Tanaka hides in town 5, then the friends could block roads $(1,2),(1,3),(1,4)$, as shown below:



Likewise, they could have $k = 3, k = 4$, or even $k = 5$ friends. On the other hand, they cannot have $k = 2$ friends: if Lulu hides in town 1 and Tanaka in town 5, then only 2 friends cannot block roads so as to stop Lulu from being able to reach Tanaka.

Each town has a different overall visibility (some towns are at a higher altitude, and thus can be seen from more places). Let $v_1, v_2, \dots, v_n$ be the visibilities of the towns.

Being very dedicated to hide-and-seek, Lulu has already observed a typical day in the INFO(1)CUP KINGDOM, and knows that during the day the visibility of an intersection may change (due to the position of the sun, traffic, etc.). More precisely, Lulu wants to process q different events. Each event affects a certain town. If an event affects town i, then v_i, the visibility of town i, becomes

$$\left(v_i + \sum_{j \in neighbours(i)} v_j \right) \bmod 998244353.$$

For every event, he is interested in the value of v_i **after the operation**.

Interaction Protocol

Your code must `include` the `"hideandseek.h"` header.

```
#include "hideandseek.h"
```

It must also implement the following functions:

```
void init(int n, int m, int k, int q, int v[], int X[], int Y[]);
int query(int i);
```

The function `init` is given n (the number of towns), m (the number of roads), k (the number of friends Lulu and Tanaka have), q (the number of events), an array v, where $v[i]$ is the initial visibility of town i, for $1 \le i \le n$, and two other arrays X and Y, denoting that there's an edge between $X[i]$ and $Y[i]$, for $1 \le i \le m$. This function is called by the committee's code exactly once, before any other calls to `query`.

The function `query` is given i, denoting that there's an query on the value of town i, described by the formula above. The function should return the value of $v[i]$ after the query. The contestant should not implement the `main` function. This will be implemented in the committee's `grader.cpp` file; you will be given a sample `grader.cpp` in the attachments. You will also be given a template `hideandseek.cpp` with a sample incomplete implementation.

Our `main` function will read from the first line of standard input integers n, m and k. The second line contains the sequence of n initial values. The next m lines each contain a pair of integers u, v, denoting that there's a two-way road between towns u and v. Line $m + 3$ contains an integer q. The next line contains q numbers, each denoting an event.

Note that the contestant is allowed to use global variables, additional functions, methods and classes.

Restrictions

- $1 \le n \le 5 \cdot 10^5$
- $1 \le m \le 6 \cdot 10^5$
- $1 \le q \le 2,5 \cdot 10^6$
- $1 \le k \le 5$
- $0 \le v_i < 998244353$

#	Points	Restrictions
1	23	$1 \leq n, q \leq 2000$
2	21	$1 \leq n, m, q \leq 2 \cdot 10^5$
3	25	$k = 1$
4	31	No further restrictions.

Examples

Input file	Output file	Explanations

5 6 3
1 2 3 4 5
1 2
1 3
1 4
2 5
3 5
4 5
3
1 5 3
	10
	14
	27
		The network of towns is the one represented in the diagram below. After the first query, the visibility of town 1 becomes 1 + 2 + 3 + 4 = 10. After the second query, the visibility of town 5 becomes 5 + 2 + 3 + 4 = 14. After the third query, the visibility of town 3 becomes 3 + 10 + 14 = 27.