CP1 [2023] Endsem Lab

A. Best Friends Forever?

1 second, 1024 megabytes

A group of **3 friends** is considered very interesting. They can even rule a whole country or a Mafia with a strong bond. But it is also considered very delicate, as unequal distribution of affection can make one feel left out and lonely.

Our mission is to educate such groups to avoid this, but we must find such a group of friends.

People are denoted with *integer IDs*.

You are given:

- **N** -> Total number of people
- **M pairs of *ID*** -> Each pair denoting friendship between **Person A** and **Person B**

You have to print all such **DISTINCT** groups, each sorted by their *IDs*.

`DISTINCT GROUPS:` *[1,2,3] [1,2,4]*

`NON-DISTINCT GROUPS:` *[1,2,3] [2,3,1]*

CONSTRAINTS

$1 \leq N \leq 100$

$1 \leq M \leq (N * (N-1)/2)$

$1 \leq ID \leq N$

Input
First line contains 2 integers **N** and **M**

Next **M** lines contain 2 integers **A** and **B** which denotes friendship between **Person A** and **Person B**.

Output
Print all the **DISTINCT** groups of 3 friends, each sorted by their *IDs*.

If **NO** such group exists, print **-1**

input
6 6 3 1 3 2 1 2 5 4 6 5 4 6
output
1 2 3 4 5 6

input
3 3 1 1 2 2 3 3
output
-1

input
6 8 3 1 3 2 1 2 5 4 6 5 4 6 1 4 4 2
output
1 2 3 1 2 4 4 5 6

B. Another Reordering Problems

1 second, 256 megabytes

You have 2 strings, S and T, and you want to modify S such that it is equal to T. To do so, you can perform 0 or more operations, and in each operation you can remove the first character from S and insert it back into S at any index.

Find the minimum number of operations to make S equal to T.

Input
Each testcase has 3 lines.

The first line contains $n (1 \leq n \leq 2 \times 10^4)$, the length of the strings. The next 2 lines contain the strings S and T. The strings only contain lower case alphabets.

Output
Print the minimum number of operations required. If it is impossible, then output -1.

input
4 abab abba
output
2

We can convert S into T using 2 operations as follows:

abab $\rightarrow$ baba (**I**nsert at index 3)

baba $\rightarrow$ abba (**I**nsert at **I**ndex 2)

C. Handshakes That Don't Cross

1 second, 256 megabytes

You are given an even number of people n that stand around a circle and each person shakes hands with someone else so that there are $n/2$ handshakes total.

Output the number of ways these handshakes could occur such that none of the handshakes cross.

Since the answer could be very large, return it modulo $10^9 + 7$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(2 \leq t \leq 100)$. The description of the test cases follows.

The only line of each test case contains a single integer n $(2 \leq n \leq 2^* 10^3)$.

It is guaranteed that the sum of n across all test cases does not exceed 2000.

Output

Output the number of ways these handshakes could occur such that none of the handshakes cross.

input
2 4 6
output
2 5

Explanation of the first test case: There are two ways to do it, the first way is [(1,2),(3,4)] and the second one is [(2,3),(4,1)].

Refer this (http://shorturl.at/oEFNR) link for n = 6.

D. XOR on trees

0.5 seconds, 256 megabytes

You are given a rooted tree with n nodes labeled 1 to n rooted at 1, with each node having the weight a_i. You have to answer q queries about the tree. A query of type 1 will be of the format $1\ n\ w$, where you have to change the weight of node n to w. A query of type 2 will be of the format $2\ n$, where you have to give the value of the bitwise XOR of the weight of all nodes, except for the nodes belonging to the subtree rooted at n. You can assume the XOR of an empty set to be 0.

Input

The first line consists of two integers n, q. $(1 \leq n, q \leq 10^5)$

The second line consists of n integers a_i, the initial weights of the nodes. $(1 \leq a_i \leq 10^9)$

The next $n-1$ lines consist of two integers x and y, representing an edge between x and y. $(1 \leq x, y \leq n)$

The next q lines consist of queries either of type 1 or 2.

Output

Print 1 line for every query of type 2 containing a single integer.

input
7 4 1 2 3 4 5 6 7 1 2 2 4 2 5 1 3 3 6 3 7 2 1 2 2 1 4 0 2 2
output
0 3 3

E. Eccentric Wizard

1 second, 256 megabytes

You are travelling to lands of Celestoria to meet a wizard and purchase k ancient artifacts you desperately need. On your way there you come across another returning traveler who tells you how he got robbed by the wizard as he kept changing prices of remaining artifacts everytime he made an purchase.

Upon further inquiry, you learn that the wizard adds a markup or discount on artifacts based on the previous artifact purchased. You manage to find the base price of each of the n artifacts the wizard has and also obtain a $n \times n$ matrix $disc$ where $disc_{x,y}$ indicates discount on x^{th} artifact if you had previously purchased artifact y.

A negative value of $disc_{x,y}$ indicates a markup. If the price p of artifact is negative after applying the discount, the wizard will pay you $abs(p)$ for buying the artifact.

Since these are very rare artifacts, even the wizard only possesses single specimen of every artifact. Can you find the minimum price required to obtain the k artifacts you need ?

Input

The first line of each testcase contains 2 integers, $n(1 \leq n \leq 14)$, the number of artifacts the wizard has, and $k(1 \leq k \leq n)$, the number of artifacts you want.

The next line contains an array $a(1 \leq a_i \leq n\ \forall i \in [1, k])$ of length k, a_i indicating that you need the i^{th} artifact.

The next line contains an array p of n integers, p_i denoting the base price of i^{th} artifact $(1 \leq p_i \leq 10^9)$.

The next n lines contain the matrix $disc$ with n integers in each line. The x^{th} row in the matrix has n integers denoting the discount available on artifact x for buying each of the n artifacts immediately before $(-10^9 \leq disc_{x,y} \leq 10^9)$.

Output

Print the minimum price you need to pay to buy the all the k artifacts. Incase you are able to earn money, output the profit as negative value.

input
3 2 1 2 1 2 3 0 0 0 0 0 0 0 0 0
output
3

input
3 2 1 2 1 2 3 0 -1 0 0 0 10 2 0 0
output
-6

input
5 2 4 5 1 10000 100 10 10 0 0 0 0 0 0 0 0 0 0 1000 0 0 0 0 0 0 0 0 0 0 0 0 0 0

output
-879

In the first case, the order in which you purchase the artifacts does not make any difference.

In the second case, if we purchase artifact 1 and then 2, it costs 3, and purchasing artifact 2 and then 1 costs 4 units due to markup of 1. But if we buy the artifact 3, which we didn't even need, in order $1 \to 3 \to 2$, then we can make a profit of 6.

In the third test, purchasing artifacts 1 and 3 (even though they are not needed) leads to profit of 899, after which we can purchase 4 and 5 artifacts afterwards.