Bangladesh Association of Problem Setters

Problem A

K^{th} Permutation Revisited

A sequence of **N** integers is called a permutation if it contains all integers from **1 to N** exactly once. For example: if **N=3**, then the set is **{1,2,3}** and it has a total **3! (= 6)** permutations.
By listing and labeling, all of the permutations in order are shown below.

1. {1, 2, 3}
2. {1, 3, 2}
3. {2, 1, 3}
4. {2, 3, 1}
5. {3, 1, 2}
6. {3, 2, 1}

In this problem, you are given **N**, **K** and **M**. You have to find the summation of the last **M** integers (rightmost) of the **K^{th}** permutation of **N** integers.

For example, if **N=3, K=2 and M=2**, then the K^{th} Permutation is **{1, 3, 2}**. So, the answer should be **3+2=5**.

Input

The first line will contain **T ($1 \le T \le 100000$)**, the number of test cases. Each test case consists of one line containing three integers **N ($1 \le N \le 10^8$), K ($1 \le K \le \min(10, N!)$) and M ($1 \le M \le N$).**

Output

For each case, print one line with "**Case <x>: <y>**", where **x** is the case number and **y** is the summation of last **M** integers (rightmost) of the **K^{th}** permutation of the sequence consisting of **N** integers. Please see the sample IO for more details.

Sample Input	Output for Sample Input
4 3 2 1 3 2 2 4 1 4 4 1 2	Case 1: 2 Case 2: 5 Case 3: 10 Case 4: 7

0 1 2 3

3

0 1 3 2

easy

Problem B

Yet Another Suffix Array Problem

Stelle has recently learned the Suffix Array algorithm. But she got upset realizing that problems which require Suffix Array rarely come up in Bangladeshi contests! So, the problem setter has decided to develop a problem and give it to her to solve it!

A sequence of N integers is called a permutation if it contains all integers from 1 to N exactly once. The problem setter will give Stelle a permutation P. Then he will ask her **Q** queries, each being like this: "What is the lexicographically smallest **K**th subarray of the permutation **P**?"

Input

The first line will contain **T (1 ≤ T ≤ 10)**, the number of test cases. The first line of each test case will contain two integers **N (1 ≤ N ≤ 100000)** and **Q (1 ≤ Q ≤ 100000)**, the size of the array, and the number of queries respectively. The next line contains **N** space-separated integers, denoting the elements of the array. The next line contains **Q** space-separated integers, each denoting the query **K** ($1 \le K \le \frac{N \times (N+1)}{2}$).

Output

For each of the queries, print a line containing two space-separated integers **L R** such that **1 ≤ L ≤ R ≤ N** and **P[L...R]** is the **K**th lexicographically smallest subarray of **P**.

Sample Input	Output for Sample Input
1 3 2 2 3 1 1 6	3 3 2 3

The subarrays of [2 3 1] sorted lexicographically smallest to largest are:

[1]
[2]
[2 3]
[2 3 1]
[3]
[3 1]

1 + 3 + 2

a + idx

1 - a[i]

a
1
4
6
9

3

AC

Problem C

Connect the Cities

The King of Merryland recently conquered a lot of cities to show the strength of his empire. But the newly conquered cities have no roads between them to easily communicate. As a result, it will be hard for the King to rule and develop the business. So, he asked one of his ministers to build bidirectional roads among them but he also gave the following conditions:

- After the roads are built, all the cities should be easily accessible. From any city, they should be able to travel anywhere using **at most two roads**. A single road connects two cities. Also, there will be only a single way to reach from one city to another.
- As the king is also concerned about finance, he wants the total cost of building the roads to be minimum. Each proposed road has a cost associated with it.

The minister has all the manpower to build the roads and he has also prepared a list of roads that can be built. But he has no idea how to choose the roads from that proposed list so that the cost will be minimal. Someone told the minister about you and your friends who are experts in programming and solving such problems. So, the minister is now asking you to help him out.

Input

The first line will contain **T** $(1 \le T \le 10)$, the number of test cases. Each case will start with two integers **N** $(1 \le N \le 100000)$ & **M** $(N-1 \le M \le 300000)$ where **N** represents the number of new cities the King has conquered and **M** represents the number of proposed roads. The cities are named with integers from **1 to N**. Each of the next **M** lines will contain three integers, **U** $(1 \le U \le N)$, **V** $(1 \le V \le N)$ & **W** $(1 \le W \le 1000000000)$ while $U \ne V$. Here **U** & **V** represents the name of two cities and **W** is the cost of building the bidirectional road between them.

Output

You will need to print the minimum cost to build the roads so that the King's conditions are fulfilled in the format: "**Case X: Y**" (without quotes), where **X** is the case number and **Y** is the minimum. If the proposed roads cannot fulfill the King's conditions, then simply print **"Case X: Impossible"** (without quotes) where **X** is the case number.

Sample Input	Output for Sample Input
3 2 1 1 2 1 3 3 1 2 1 2 3 2 3 1 1 4 3 1 2 1 2 3 1 3 4 1	Case 1: 1 Case 2: 2 Case 3: Impossible

Bangladesh Association of Problem Setters

Problem D

Nim Heaps

In the game of Nim, there are several heaps of stones. Two players make moves alternately. On each turn, the current player selects any heap and takes some positive number of stones from it. The player who takes the last stone wins the game.

Alice and Bob love playing the Nim game. They play it so often that they learned how to determine the winner at a first glance: if there are $a_1, a_2, a_3 \cdots a_n$ stones in the heaps, the first player wins if and only if the bitwise xor $a_1 \oplus a_2 \oplus \cdots \oplus a_n$ is nonzero.

They got so bored playing regular nim, that they decided to come up with a variation. Before starting the game, Alice and Bob agree upon two integers n and k. Then the game proceeds as follows -

- First Alice makes n heaps of stones. She can choose the number of stones in each heap. Each heap should have a positive number of stones not exceeding $2^k - 1$. Furthermore, the number of stones in the heaps should be in non-decreasing order. Formally, If Alice chooses $a_1, a_2, a_3 \cdots a_n$ as the number of stones in the heaps, then

$$1 \le a_1 \le a_2 \le \cdots \le a_n \le 2^k - 1$$

- Then Bob chooses a nonempty subset of heaps and removes the others.
- Then the regular Nim game starts on the chosen heaps with Alice to move first.

Alice wants to know how many ways she can choose the number of stones in the heaps such that no matter how Bob responds, she will always win. Since this number can be large she is happy to know its value **modulo 998244353.**

Input

The first line will contain a single integer $T\ (1 \le T \le 10^5)$ denoting the number of test cases.
Each test case consists of one line containing two integers n and $k\ (1 \le n, k \le 10^6)$.

Output

For each test case, print a single integer in a new line - the number of ways Alice can choose the heaps **modulo 998244353.**

Sample Input	Output for Sample Input
3 1 3 2 2 3 2	7 3 0

Explanation

In the first case, there is only one heap and Alice can choose anywhere between 1 to 7 stones for that heap.

In the second case, there are 2 heaps and each heap can have at most 3 stones. The two heaps cannot have the same number of stones, otherwise Bob can choose the entire set and win. Considering this, there are 3 ways to assign the number of stones in the heaps - (1, 2), (1, 3), (2, 3).

In the third case, there are 3 heaps and each heap can have at most 3 stones. No two heaps can have the same number of stones, otherwise Bob can pick those two heaps only and win. Therefore, the number of stones must be 1, 2, 3. But then the entire set of heaps has a xor-sum of zero and Bob can pick the entire set. Thus Alice has no way of assigning the heaps so that she wins.

Problem E

Min Cost Sort

DP

You are given a permutation $a(a_1,...,a_n)$. You want to sort it. You can make 0 or more operations, in one operation you can choose any two indices **i** and **j** such that $1 \le i < j \le n$ and swap a_i and a_j with cost **i+j**. The total cost to sort the permutation is the sum of the cost of all the operations you make. You have to output the minimum possible cost to sort the permutation.

A permutation of size **n** is a sequence of **n** integers such that every integer from **1** to **n** exists exactly once in the sequence.

Input

The first line will contain a single integer **T (1 ≤ T ≤ 5)** denoting the number of test cases. Each test case will start with a line having a single integer **N (1 ≤ N ≤ 300)**, denoting the size of the permutation. The following line will contain **N** integers denoting the permutation.

Output

For each test case, print a single integer, the minimum cost to sort the permutation.

Sample Input	Output for Sample Input
4 3 1 2 3 3 3 2 1 3 2 3 1 3 1 3 2	0 4 7 5

Explanation:

Explanation for 3rd case: Initial permutation is,

2 3 1

We make one swap between 1st and 2nd index with cost (1+2). Now the permutation is,

3 2 1

We again make a swap between, 1st and 3rd index with cost (1+3) Now the permutation is,

1 2 3

Total Cost is (1+2) + (1+3) = 7

Problem F

Power Sequence

Bangladesh Association of Problem Setters

NT

You are given an integer **N** and an array of **N** integers, $B_1, B_2, B_3, \dots B_N$.

A sequence of integers $X = [X_1, X_2, X_3, \dots X_k]$ is considered valid if

- Every integer is between 1 and N. Formally, $1 \le X_i \le N$ for all i.
- Their sum is equal to **N**. Formally, $X_1 + X_2 + X_3 + \dots + X_k = N$.

OR bitmask

For a valid sequence $X = [X_1, X_2, X_3, \dots X_k]$, the beauty of the sequence is calculated as

$$F(X) = B_{X_1} + B_{X_1}B_{X_2} + B_{X_1}B_{X_2}B_{X_3} + B_{X_1}B_{X_2}B_{X_3}B_{X_4} + \dots + B_{X_1}B_{X_2}B_{X_3}..B_{X_k}$$

More formally, $F(X) = \sum_{i=1}^{k} \prod_{j=1}^{i} B_{X_j}$

You need to find the sum of the **M-th** power of the function **F** for all possible valid sequences. More formally

you need to find, $\sum_{S \in \{all\ possible\ valid\ sequences\}} F(S)^m$

60

As the answer can be very large print it modulo **998244353**.

Input

First line will contain an integer **T(1 ≤ T ≤ 20)** which represents the number of test cases. For each test case there will be two lines of inputs. First line will contain two space separated integers **N(1 ≤ N ≤ 32000)** and **M(0 ≤ M ≤ 20)**. Second line will contain **N** space separated integers where the i-th integer will represent B_i ($1 \le B_i \le 10^5$).

You may safely assume that, sum of N for all over test cases does not exceed 32000.

Output

For each case, print one line with "**Case <x>: <y>**", where **x** is the case number and **y** is the answer modulo **998244353**. Please see the sample output for more details.

Sample Input	Output for Sample Input
2 3 1 2 4 1 3 2 2 4 1	Case 1: 37 Case 2: 441

Explanation:

here for the second test case, N = 3 and M = 2.

You can make 4 valid sequences. They are [3], [1, 2], [2, 1], [1, 1, 1].

- $F([3])^2 = (B_3)^m = 1^2 = 1$
- $F([1, 2])^2 = (B_1 + B_1 * B_2)^m = (2 + 2 * 4)^2 = 10^2 = 100$
- $F([2, 1])^2 = (B_2 + B_2 * B_1)^m = (4 + 4 * 2)^2 = 12^2 = 144$
- $F([1, 1, 1])^2 = (B_1 + B_1 * B_1 + B_1 * B_1 * B_1)^m = (2 + 2 * 2 + 2 * 2 * 2)^2 = 14^2 = 196$

answer = 1 + 100 + 144 + 196 = 441

Problem G

Creative Kindergarten

Sharif, a 4 years old kid who lived in Dumbland, was very eager to know new things and always kept asking a lot of questions. One day his parents decided to admit him to the best school in the county. So, they admitted Sharif to a school named "Creative Kindergarten", the best institute for the kids in Dumbland.

Sharif was really excited after getting admission. In his very first class, a math teacher came into the class and wrote "**2, 3, 5, 7, 11, 13, 17 and 19 are prime numbers in between 1 to 20**".

Sharif asked - "What is a prime number Sir?" Teacher replied-" You asked a question! Just memorize it."

In the next class, the teacher asked Sharif to list out **any K prime** numbers between 1 to 20. In this problem, you need to verify Sharif's answer. Sharif's answer will be considered correct if he can list out **K distinct prime numbers between 1 to 20 in any order**.

Input

The first line will contain **T (1 ≤ T ≤ 1000)**, the number of test cases. The first line of each test case will contain an integer **K (1 ≤ K ≤ 8)**. The next line contains **K** space-separated distinct integers, denoting Sharif's answer.

Output

For each case, print one line starting with "**Case <x>:** ", where **x** is the case number. If Sharif's answer is correct, then print "**Yes**", Otherwise Print "**No**".

Sample Input	Output for Sample Input
3 4 2 3 5 7 5 3 5 2 17 11 5 3 5 2 7 4	Case 1: Yes Case 2: Yes Case 3: No

Graph

Problem H

Is this Graph?

We know you all love short problem statements, so here's a treat for you.

You are given an undirected, unweighted graph of N nodes with M edges. Then you have to process Q operations of two types, each of which is an update or a query. They will follow the format described below.

a. Update: **1 U V**
For each update, you will add a new edge connecting nodes U and V.

b. Query: **2 U V**
For each query, you need to tell how many edges are there in the graph such that, if you remove any of those, then U and V will be disconnected. If **U** and **V** are already disconnected, output -1.

Input

Input starts with an integer **T**, the number of test cases. The first line of each test case starts with two integers **N** and **M**, indicating the number of nodes and edges of the graph respectively. Then each of the next **M** lines will contain two integers **U** and **V**, meaning there's an edge between these two nodes. In the next line, there will be one Integer **Q**, the number of operations. Each of the next **Q** lines will describe either an update or query which will follow the format described above. There won't be any duplicate edges.

Constraint

- **$1 \le T \le 100$**
- **$2 \le N \le 100000$**
- **$1 \le M \le min(100000, \frac{N \times (N-1)}{2})$**
- **$1 \le U, V \le N,\ U \ne V$**
- **$1 \le Q \le 100000$**
- **The summation of all N, M, and Q in a test file will be less than or equal to 300000.**

Output

Print the case number in a single line. Then for every query, you need to print the answer to that query in each line.

Sample Input	Output for Sample Input
1 6 4 1 2 2 3 4 5 5 6 7 2 1 2 2 1 3 2 1 4 1 1 3 2 1 3 1 3 4 2 1 6	Case 1: 1 2 -1 0 3

Problem I

Beautiful Blocks (Easy)

The only differences between the easy and the hard version are the board size (N), the number of boards (K), and the sum of N x K over all test cases.

Alice likes playing a game on an **N x N (N is even)** board where each cell contains some points. Rows are numbered from **1** to **N** from the top to the bottom, and columns are numbered from **1** to **N** from the left to the right. She starts from the **(1, 1)** cell and ends her journey on the **(N, N)** cell. But on each move, she can only move in the right or down cell. When she is on a cell, she collects all the points in that cell. Alice loves points a lot. So she always moves in such a way that the total number of points she can collect is maximized.

Bob, the magician, has **K empty N x N** board. He also has $\frac{KN^2}{4}$ small **(2 x 2)** blocks where each cell of the blocks contains some points. He can magically shuffle the cells of any block. After the shuffling, he fills the empty boards with blocks in any way he wishes and gives the filled boards to Alice. **Alice then plays her game on all of the boards.** As Bob is a kind magician, he wants to make Alice happy as much as possible. So, he shuffles the cells of the blocks and fills the empty boards with the blocks in such a way that Alice can get the maximum number of points. Can you find out how many points Alice will get?

Input

The first line will contain a single integer **T (1 ≤ T ≤ 300)** denoting the number of test cases. For each test case, the first line will have two space-separated integers **N (2 ≤ N ≤ 300 and N is even)** and **K (K = 1)**, denoting the dimension of the empty boards and the number of boards respectively. The description of $\frac{KN^2}{4}$ small **(2 x 2)** blocks follows in the next $\frac{KN^2}{2}$ lines. Each block is represented in two lines, where each line contains two integers denoting the points (**0 ≤ points in each cell ≤ 10^9**) in the cells of this block. **The sum of N * K over all test cases does not exceed 600.**

Output

For each test case, print the maximum number of points Alice will get in a single line. Please see the sample for details.

Sample Input

```
2
2 1
1 4
2 3
4 1
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
```

Output for Sample Input

```
9
86
```

Explanation:

For the first case:

There is only one block,

1	4
2	3

After Bob arranges points in the block and fills the empty board, the board looks like this:

4	1
3	**2**

The cells in bold represent the path followed by Alice.
So the maximum number of points obtained by Alice = 4 + 3 + 2 = 9.

For the second case:

There are four blocks,

1	2
3	4

5	6
7	8

9	10
11	12

13	14
15	16

After shuffling the cells in the blocks, the four blocks look like this:

1	2
3	4

6	5
8	7

12	10
9	11

16	13
15	14

Then Bob fills the empty board as the following,

16	13	6	5
15	**14**	**8**	7
1	2	**12**	**10**
3	4	9	**11**

The cells in bold represent the path followed by Alice.
So the maximum number of points obtained by Alice = 16 + 15 + 14 + 8 + 12 + 10 + 11 = 86.

Problem J

Beautiful Blocks (Hard)

The only differences between the easy and the hard version are the board size (N), the number of boards (K), and the sum of N x K over all test cases.

Alice likes playing a game on an **N x N (N is even)** board where each cell contains some points. Rows are numbered from **1** to **N** from the top to the bottom, and columns are numbered from **1** to **N** from the left to the right. She starts from the **(1, 1)** cell and ends her journey on the **(N, N)** cell. But on each move, she can only move in the right or down cell. When she is on a cell, she collects all the points in that cell. Alice loves points a lot. So she always moves in such a way that the total number of points she can collect is maximized.

Bob, the magician, has **K empty N x N** board. He also has $\frac{KN^2}{4}$ small **(2 x 2)** blocks where each cell of the blocks contains some points. He can magically shuffle the cells of any block. After the shuffling, he fills the empty boards with blocks in any way he wishes and gives the filled boards to Alice. **Alice then plays her game on all of the boards.** As Bob is a kind magician, he wants to make Alice happy as much as possible. So, he shuffles the cells of the blocks and fills the empty boards with the blocks in such a way that Alice can get the maximum number of points. Can you find out how many points Alice will get?

Input

The first line will contain a single integer **T (1 ≤ T ≤ 1000)** denoting the number of test cases. For each test case, the first line will have two space-separated integers **N (2 ≤ N ≤ 1000 and N is even)** and **K (1 ≤ K ≤ 1000)**, denoting the dimension of the empty boards and the number of boards respectively. The description of $\frac{KN^2}{4}$ small **(2 x 2)** blocks follows in the next $\frac{KN^2}{2}$ lines. Each block is represented in two lines, where each line contains two integers denoting the points (**0 ≤ points in each cell ≤ 10^9**) in the cells of this block.

The sum of N * K over all test cases does not exceed 2000.

Output

For each test case, print the maximum number of points Alice will get in a single line. Please see the sample for details.

Sample Input

```
3
2 1
1 4
2 3
4 1
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
2 2
1 2
3 4
1 2
3 4
```

Output for Sample Input

```
9
86
18
```

Explanation:

For the first case:

There is only one block,

1	4
2	3

After Bob arranges points in the block and fills the empty board, the board looks like this:

4	1
3	**2**

The cells in bold represent the path followed by Alice.

So the maximum number of points obtained by Alice = 4 + 3 + 2 = 9.

For the second case:

There are four blocks,

1	2
3	4

5	6
7	8

9	10
11	12

13	14
15	16

After shuffling the cells in the blocks, the four blocks look like this:

1	2
3	4

6	5
8	7

12	10
9	11

16	13
15	14

Then Bob fills the empty board as the following,

16	13	6	5
15	**14**	**8**	7
1	2	**12**	**10**
3	4	9	**11**

The cells in bold represent the path followed by Alice.

So the maximum number of points obtained by Alice = 16 + 15 + 14 + 8 + 12 + 10 + 11 = 86.

For the third case:

The filled boards look like this:

2	1
3	**4**

2	1
3	**4**

So the maximum number of points obtained by Alice = 2 + 3 + 4 + 2 + 3 + 4 = 18.