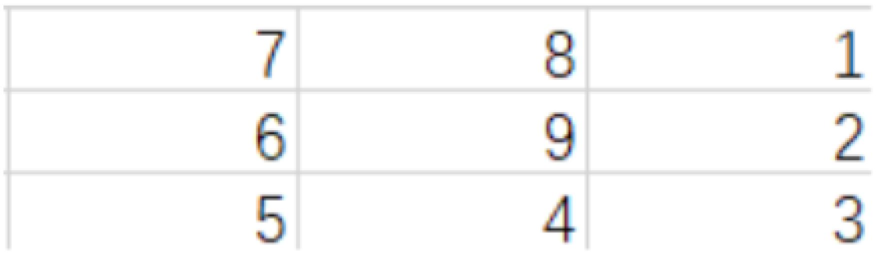
The Preliminary Contest for ICPC Asia Nanjing 2019

The beautiful values of the palace

限制

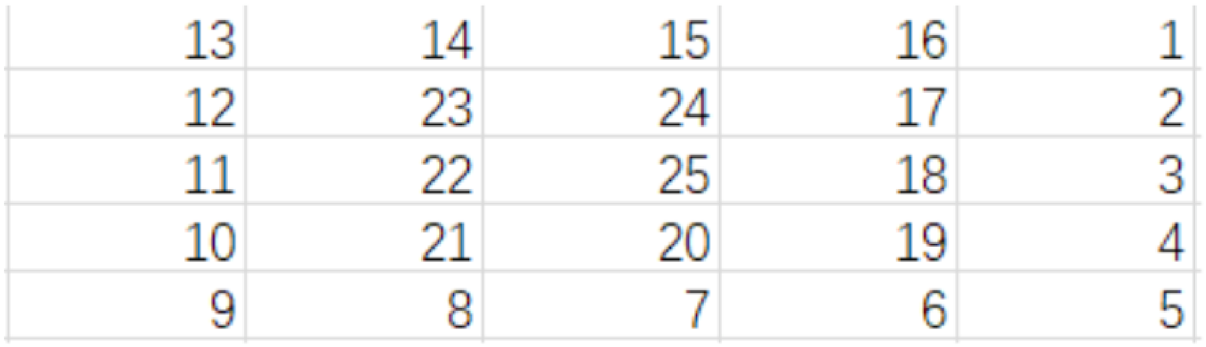
- 1500 ms
- 512 MB

Here is a square matrix of $n * n$, each lattice has its value (n must be odd), and the center value is $n * n$. Its spiral decline along the center of the square matrix (the way of spiral decline is shown in the following figure:)



7	8	1
6	9	2
5	4	3

square matrix of 3*3



13	14	15	16	1
12	23	24	17	2
11	22	25	18	3
10	21	20	19	4
9	8	7	6	5

square matrix of 5*5

The grid in the lower left corner is (1,1) and the grid in the upper right corner is (n , n)

Now I can choose m squares to build palaces, The beauty of each palace is equal to the digital sum of the value of the land which it is located. Such as (the land value is 123213,the beautiful values of the palace located on it is $1+2+3+2+1+3=12$) (666 -> 18) (456 ->15)

Next, we ask p times to the sum of the beautiful values of the palace in the matrix where the lower left grid(x_1, y_1), the upper right square (x_2, y_2).

Input

The first line has only one number T .Representing T-group of test data $(T \leq 5)$

The next line is three number: n m p

The m lines follow, each line contains two integers the square of the palace (x, y)

The p lines follow, each line contains four integers : the lower left grid (x_1, y_1) the upper right square (x_2, y_2)

Output

Next, $p_1 + p_2 ... + p_T$ lines: Represent the answer in turn
$(n \leq 10^6)(m, p \leq 10^5)$

Sample Input

```
1
3 4 4
1 1
2 2
3 3
2 3
1 1 1 1
2 2 3 3
1 1 3 3
1 2 2 3
```

Sample Output

```
5
18
23
17
```

super_log

In Complexity theory, some functions are nearly $O(1)$, but it is greater then $O(1)$. For example, the complexity of a typical disjoint set is $O(n\alpha(n))$. Here $\alpha(n)$ is Inverse Ackermann Function, which growth speed is very slow. So in practical application, we often assume $\alpha(n) \le 4$.

However $O(\alpha(n))$ is greater than $O(1)$, that means if n is large enough, $\alpha(n)$ can greater than any constant value.

Now your task is let another slowly function $log*\ x$ reach a constant value b. Here $log*$ is iterated logarithm function, it means "the number of times the logarithm function iteratively applied on x before the result is less than logarithm base a".

Formally, consider a iterated logarithm function log_a^*

$$\log_a^*(x) = \begin{cases} -1 \ \ if\ x < 1 \\ 1 + \log_a^*(\log_a x) \ \ if\ x \ge 1 \end{cases}$$

Find the minimum positive integer argument x, let $log_a^*(x) \ge b$. The answer may be very large, so just print the result x after mod m.

Input

The first line of the input is a single integer $T(T \le 300)$ indicating the number of test cases.

Each of the following lines contains 3 integers a , b and m.

$2 \le a \le 1000000$

$0 \le b \le 1000000$

$1 \le m \le 1000000$

Output

For each test case, output x mod m in a single line.

Hint

In the $4-th$ query, $a=3$ and $b=2$. Then $log_3^*(27)=1+log_3^*(3)=2+log_3^*(1)=3+(-1)=2\geq b$, so the output is 27 mod $16=11$.

Sample Input

```
5
2 0 3
3 1 2
3 1 100
3 2 16
5 3 233
```

Sample Output

```
1
1
3
11
223
```

Tsy's number 5

限制

- 1500 ms
- 256 MB

Tsy is a math enthusiast and his favorite thing is to get the sum of function. One day he was studying the Euler function, while writing a formula

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\varphi(i)\varphi(j)2^{\varphi(i)\varphi(j)}$$

Tsy took it to his friend Zxy for solution. But Zxy don't know any math at all. As the friends of Zxy, can you help him ?

The result need to 998244353 modulo.

Input

The first line of the input contains an integer $T(1 \le T \le 5)$,denoting the number of test cases.

A single number $n(n \le 10^5)$

Output

For each test case,print a single line containing an integer,denoting the result.

Sample Input

```
1
3
```

Sample Output

```
104
```

Robots

限制

- 1000 ms
- 256 MB

Given a directed graph with no loops which starts at node 1 and ends at node n.

There is a robot who starts at 1, and will go to one of adjacent nodes or stand still with **equal probability** every day. Every day the robot will have durability consumption which equals to the number of passed days.

Please calculate the expected durability consumption when the robot arrives at node n.

It is guaranteed that there is **only one** node (node 1) whose in-degree is equal to 0, and there is **only one** node (node n) whose out-degree is equal to 0. And there are no multiple edges in the graph.

Input

The first line contains one integer $T(1 \le T \le 10)$

For each case,the first line contains two integers $n(2 \le n \le 10^5)$ and $m(1 \le m \le 2 \times 10^5)$, the number of nodes and the number of edges, respectively.

Each of the next m lines contains two integers u and v $(1 \le u, v \le n)$ denoting a directed edge from u to v.

It is guarenteed that $\sum n \le 4 \times 10^5$, and $\sum m \le 5 \times 10^5$.

Output

Output T lines.Each line have a number denoting the expected durability consumption when the robot arrives at node n.

Please keep **two decimal places**.

Sample Input

```
1
5 6
1 2
2 5
1 5
1 3
3 4
4 5
```

Sample Output

```
9.78
```

K Sum

限制

- 3000 ms
- 256 MB

Define function

$$f_n(k) = \sum_{l_1=1}^{n} \sum_{l_2=1}^{n} \dots \sum_{l_k=1}^{n} (\gcd(l_1, l_2, \dots, l_k))^2$$

.

Given $n(1 \le n \le 10^9), k(2 \le k \le 10^{10^5})$, please calculate

$$\sum_{i=2}^{k} f_n(i)$$

modulo $10^9 + 7$.

Input

There are multiple test cases, the first line contains an integer $T(1 \le T \le 10)$, denoting the number of test cases.

For the next T lines, each line contains two integers $n(1 \le n \le 10^9)$ and $k(2 \le k \le 10^{10^5})$.

Output

For each test case, print one line contains an integer, which is the value of $\sum_{i=2}^{k} f_n(i)$.

Sample Input

```
2
2 2
100 3
```

Sample Output

```
7
4331084
```

Greedy Sequence

限制

- 5000 ms
- 256 MB

You're given a permutation a of length n ($1 \le n \le 10^5$).

For each $i \in [1, n]$, construct a sequence s_i by the following rules:

1. $s_i[1] = i$;
2. The length of s_i is n, and for each $j \in [2, n]$, $s_i[j] \le s_i[j-1]$;
3. First, we must choose all the possible elements of s_i from permutation a. If the index of $s_i[j]$ in permutation a is $pos[j]$, for each $j \ge 2$, $|pos[j] - pos[j-1]| \le k$ ($1 \le k \le 10^5$). And for each s_i, every element of s_i must occur in a **at most once**.
4. After we choose all possible elements for s_i, if the length of s_i is smaller than n, the value of **every undetermined element** of s_i is 0;
5. For each s_i, we must make its weight high enough.

Consider two sequences $C = [c_1, c_2, ...c_n]$ and $D = [d_1, d_2, ..., d_n]$, we say the weight of C is **higher than** that of D if and only if there exists an integer k such that $1 \le k \le n$, $c_i = d_i$ for all $1 \le i < k$, and $c_k > d_k$.

If for each $i \in [1, n]$, $c_i = d_i$, the weight of C is equal to the weight of D.

For each $i \in [1, n]$, print the number of non-zero elements of s_i separated by a space.

It's guaranteed that there is only one possible answer.

Input

There are multiple test cases.

The first line contains one integer $T(1 \leq T \leq 20)$, denoting the number of test cases.

Each test case contains two lines, the first line contains two integers n and k ($1 \leq n, k \leq 10^5$), the second line contains n distinct integers $a_1, a_2, ..., a_n$ ($1 \leq a_i \leq n$) separated by a space, which is the permutation a.

Output

For each test case, print one line consists of n integers $|s_1|, |s_2|, ..., |s_n|$ separated by a space.

$|s_i|$ is the number of non-zero elements of sequence s_i.

There is no space at the end of the line.

Sample Input 1

```
2
3 1
3 2 1
7 2
3 1 4 6 2 5 7
```

Sample Output 1

```
1 2 3
1 1 2 3 2 3 3
```

Quadrilateral

限制

- 1000 ms
- 256 MB

A quadrilateral consists of four sides a, b, c, and d in a clockwise order, the length of each side is a positive integer and varies in a range $[L, R]$ for this side. You need to calculate the number of quadrilaterals that can be formed by a, b, c, and d. We define two quadrilaterals are different when their corresponding edges are not same. For example, we have two quadrilaterals Q1 and Q2. The edges of Q1 are a_1, b_1, c_1, and d_1 and edges of Q2 are a_2, b_2, c_2, and d_2. If $a_1 \neq a_2$ or $b_1 \neq b_2$,or $c_1 \neq c_2$ or $d_1 \neq d_2$, we call them different kinds.

Input

The first line contains an integer n, indicating the number of test cases.

The following n lines, each line contains eight numbers $x[1], x[2], \ldots, x[8]$, indicating that a ranges from $x[1]$ to $x[2]$, b ranges from $x[3]$ to $x[4]$, c ranges from $x[5]$ to $x[6]$ and d ranges from $x[7]$ to $x[8]$.
$(1 \leq N \leq 1000, 1 \leq a, b, c \leq 10^5, 1 \leq d \leq 10^3)$

Output

For each test case, print the number of quadrilaterals that can be formed.

Sample Input

```
1
1 2 1 2 1 2 1 2
```

Sample Output

```
16
```

Holy Grail

限制

- 1000 ms
- 256 MB

As the current heir of a wizarding family with a long history,unfortunately, you find yourself forced to participate in the cruel Holy Grail War which has a reincarnation of sixty years.However,fortunately,you summoned a Caster Servant with a powerful Noble Phantasm.When your servant launch her Noble Phantasm,it will construct a magic field,which is actually a **directed graph** consisting of **n vertices** and **m edges**.More specifically,the graph satisfies the following restrictions :

- Does not have multiple edges(for each pair of vertices ***x*** and ***y***, there is at most one edge between this pair of vertices in the graph) and does not have self-loops(edges connecting the vertex with itself).
- May have **negative-weighted edges**.
- Does not have a **negative-weighted loop**.
- **n<=300** , **m<=500**.

Currently,as your servant's Master,as long as you add extra **6** edges to the graph,you will beat the other 6 masters to win the Holy Grail.

However， you are subject to the following restrictions when you add the edges to the graph:

- Each time you add an edge whose cost is c,it will cost you c units of Magic Value.Therefore,you need to add an edge which has the lowest weight(it's probably that you need to add an edge which has a negative weight).
- Each time you add an edge to the graph,the graph must not have negative loops,otherwise you will be engulfed by the Holy Grail you summon.

Input

Input data contains multiple test cases. The first line of input contains integer **t** — the number of test
cases ($1 \leq t \leq 5$).

For each test case,the first line contains two integers n,m,the number of vertices in the graph, the initial number of edges in the graph.

Then m lines follow, each line contains three integers **x, y and w** ($0 \leq x, y < n$,-10^9≤w≤ 10^9, $x \neq y$) denoting an edge from vertices x to y (**0-indexed**) of weight w.

Then 6 lines follow, each line contains two integers **s,t** denoting **the starting vertex** and **the ending vertex** of the edge you need to add to the graph.

It is guaranteed that there is not an edge starting from **s to t** before you add any edges and there must exists such an edge which has the lowest weight and satisfies the above restrictions, meaning the solution absolutely exists for each query.

Output

For each test case,output 6 lines.

Each line contains the weight of the edge you add to the graph.

Sample Input

```
1
10 15
4 7 10
7 6 3
5 3 3
1 4 11
0 6 20
9 8 25
3 0 9
1 2 15
9 0 27
5 2 0
7 3 -5
1 7 21
5 0 1
9 3 16
1 8 4
4 1
0 3
6 9
2 1
8 7
0 4
```

Sample Output

```
-11
-9
-45
-15
17
7
```

Washing clothes

限制

- 2000 ms
- 512 MB

N persons are about to engage in their favorite activity doing laundry together! The i-th person will come at t_i minute. Unfortunately, there is only one washing machine, which takes x minutes to wash one load of laundry. At any moment, the machine could only be processing at most one person's clothes. But this couldn't dispel their enthusiasm for laundry! They may also choose to wash clothes by hand, which takes y minutes. Of course, everyone has hands. So they can wash clothes at the same time by hand.

For different integer x, help them to calculate the minimal moment that all N loads of clothes have been washed! Obviously, if $x > y$, it's not necessary to use washing machine. So, you only need to calculate the answer for $x \in [1, y]$.

Input

Input contains several test cases.

For each test case:

The first line contains N, y, which are both mentioned aboved.

The second line contains N integers $t_1, t_2, \cdots t_N$, indicating the coming time of each person.

Output

For each test case, you should output y integers in one line: the i-th integer means the answer when x equals i.

Constraint

$1 \le N, y \le 10^6$

$\sum N, \sum y \le 10^6$

$0 \le t_i \le 10^9$

Sample Input

```
3 3
100 10 1
```

Sample Output

```
101 102 103
```