Problem A. Cycle Count

Input file:	`cycle.in`
Output file:	`standard output`
Balloon Color:	`Pink`

El-Khadrawy is well-known across the globe for his love of graph problems, or as he calls them, "graphs". Therefore, he couldn't miss the chance of setting a graph problem for his dear participants from Egypt.

In this problem, El-Khadrawy will give you a graph of N nodes. However, this graph doesn't contain any edges yet. Next, El-Khadrawy asks you to add exactly N directed edges to this graph, **such that each node gets exactly one edge that starts from it.**

In addition, El-Khadrawy will give you an integer C. He asks you to count the number of ways to add the N edges, such that the resulting graph contains exactly C simple cycles. Recall that a simple cycle is the one that starts from some node u, and by following the edges we can reach back to u.

El-Khadrawy likes to make his problems tricky. So, for this problem, he told you that you are allowed to make self-loops. In other words, you can have an edge that starts from some node u and ends at u itself.

Since the number of ways can be very large, print it modulo $10^9 + 7$. Two ways are considered different if the first one contains at least one edge that is not presented in the other.

Input

The first line contains an integer T ($1 \leq T \leq 10^5$) denoting the number of test cases.

Each test case contains two integers N and C ($1 \leq N \leq 10^5$) ($1 \leq C \leq 100$) denoting the number of nodes, and the number of required simple cycles respectively.

Output

For each test case print a single line containing a single integer denoting the answer to the problem.

Example

cycle.in	standard output
3 1 1 3 2 100000 100	1 3 266019816

Note

Note that the number of test cases T can be large.

Problem B. K-frequency

Input file:	`frequency.in`
Output file:	`standard output`
Balloon Color:	`Yellow`

Given an array of N elements, print the minimum number of operations needed to make the array good. Or, print -1 if there is no way to make it a good array.

By our definition, an array is called good if it does not have any number with a frequency greater than K.

The only type of operations you can make is to choose a pair of indices i and j, then set either $A_i = A_j$ or $A_j = A_i$.

Input

The first line of input will be T indicating the number of test cases.

Every test case is described in two lines. The first line contains two space-separated integers N and K ($1 \leq N, K \leq 2 \times 10^5$).

The second line contains N space separated-integers - the array A ($1 \leq A_i \leq 2 \times 10^5$).

Output

For each test case print one integer, the minimum number of operations needed to make the array good. Or, print -1 if there is no way to make it a good array.

Example

frequency.in	standard output
3 5 2 1 2 2 2 3 5 2 1 2 2 2 1 4 2 3 3 3 1	1 -1 1

Problem C. Exams

Input file: exam.in
Output file: standard output
Balloon Color: Black

Resli is studying for his final exams. He finished all of his exams except for the last one.

Resli has only N hours left. He is going to write a plan for the N hours to perform well in his final exam. He can do one of two things, either study one lecture, which will take a whole hour, or eat fast food, which will take two hours.
For example, if Resli has 5 hours left, one possible schedule is $SESS$, which means he will study for one hour, eat for two hours, study for one hour, and study for one hour, for a total of 5 hours. He could also choose one of the following plans: $SSSSS$, $ESSS$, $SSES$, $SSSE$, SEE, ESE, and EES.

Resli realized that there are a lot of ways that he can schedule the studying plan. Can you help him count the number of ways?
Note that the number of ways could be very large, so you should print the remainder of dividing it by $10^9 + 7$.

Input

The first line of the input contains one integer T,$(1 \leq T \leq 10^5)$, the number of test cases.
The only line of each test case contains one integer N,$(1 \leq N \leq 10^6)$ the remaining hours for Resli's exam.

Output

For each test case, print one integer M, the number of ways to schedule a studying plan in N hours, after taking the remainder of dividing it by $10^9 + 7$.

Example

exam.in	standard output
3 1 2 3	1 2 3

Problem D. Playing With Strings

Input file:	strings.in
Output file:	standard output
Balloon Color:	Cyan

Ali is a chocolate lover. One day he was so hungry that he fell asleep. He had a very weird dream that he was at the chocolate land. Chocolate land is a land where everything is made of chocolate. So, it is the dreamland for Ali.

Ali ate a lot of chocolate, that he was full and he started to think why a lot of chocolate brands' names are palindromes.

A palindrome string is a string that reads the same backward as forward. For example, strings (*z, aaa, aba, abccba*) are palindrome but strings (*ab, abca*) are not.

We know that eating different types of chocolate makes you very active and gives you a lot of energy. Therefore, **he got a string S with distinct lower case English letters.**

He wanted to generate a string Y from S. Initially, the string Y is empty. In each step, he appends every character from S except the last i characters to the end of Y. He wanted to apply this for each i in the range $[0, n-1]$ (in the increasing order of i), where n is the length of the string.

After each append, he wants to count the number of letters you should delete from Y to make it a palindrome after rearranging its letters in any way you want. You are asked to count the sum of all the deletions you make at each string S.

Input

The first line contains T ($1 \leq T \leq 10^5$), the number of test cases.

The next T line contains string S ($1 \leq |S| \leq 26$) with distinct lowercase letters. $|S|$ denotes the length of string S.

It's guaranteed that all letters of S are distinct.

Output

Print a single integer - the sum of deletions you should make.

Example

strings.in	standard output
2 a abc	0 3

Note

abc $->$ we need to delete 2 letters to make it palindrome.

abcab $->$ we don't need to delete any letter.

abcaba $->$ we need to delete one letter.

So, the answer is 3.

Problem E. El-Khadrawy House

Input file:	`house.in`
Output file:	`standard output`
Balloon Color:	`Orange`

El-Khadrawy once went back to 1860, and gathered people who started thinking about creating problems that need to be solved. They used to gather in El-Khadrawy House. People called this group El-Khadrawy Group (El-G for short). Back then, each El-G member used to get an envelope containing N problems, numbered from 1 to N. Each problem has a difficulty rating of H_i and an importance P_i.

El-G has summoned Q of its members to take these problems, one by one. Each of the El-G members starts with the first problem and continues to solve them until they reach some problem that they don't like. Therefore, each El-G member has a rating R_j and an IQ level equal to D_j. In addition, each El-G member has a special value K_j.

The j^{th} El-G member doesn't like the i^{th} problem if $R_j \leq H_i$. When an El-G member reaches a problem that they don't like, they will stop solving this and all other problems immediately. Since El-G members believe that the number of solved problems doesn't matter, they think that their scores equals the value of $D_j \oplus P_i$, where $\oplus$ is the bitwise XOR operation.

For each El-G member j, you can choose at most K_j problems and remove them from the envelope of this member.

For each El-G member, you want to minimize the score they get. If an El-G member manages to solve all problems in their envelope, then they will gain a score equal to zero.

Note that the score for each El-G member is calculated independently. Before each El-G member takes their envelope, you should consider that all problems are available and that you can choose new K_i ones for the new El-G member.

Input

The first line contains an integer T denoting the number of test cases.

Each test case starts with a line containing N and Q ($1 \leq N, Q \leq 10^5$) denoting the number of problems and El-G members respectively.

The next N lines contain the description of the problems. Each line contains H_i and P_i ($1 \leq H_i \leq 10^5$) ($1 \leq P_i \leq 1860$) denoting the problem's difficulty rating and importance.

The next Q lines contain the description of the El-G members. Each line contains R_j, D_j and K_j ($1 \leq R_j, K_j \leq 10^5$) ($1 \leq D_j \leq 1860$) denoting the values of the j^{th} El-G member.

It's guaranteed that the sum of N over all test cases doesn't exceed 5×10^5.

Output

For each test case, Print the minimum score that can be gotten for each El-G member on a separate line.

Example

house.in	standard output
2 4 2 2 2 3 4 5 6 7 8 1 2 1 2 4 6 4 3 5 16 1 4 3 2 3 8 2 15 2 1 7 3 2 1 2	0 0 7 3 3

Problem F. Problem Setter

Input file: mracpc.in
Output file: standard output
Balloon Color: White

Becoming a problem setter is not easy. You have lots of problems to prepare, in a very limited time. So, you usually end up busy and try to come up with problems as fast as possible.

The coordinator of the contest usually tells you that if you prepare a complete problem set of R problems for them, they will give you 10 points to use for ACPC gifts.

Since 10 points are very precious, you decided to prepare this problem set. You have N problems from an old contest you prepared on Queueforces. Each problem has a type S_i and it takes A_i to be prepared.

You don't have too much time. Therefore, you want to finish as soon as possible. However, there is one thing that might hold you down. You are not that good at any topic. So, you should spend some time learning a topic before being able to use its problems inside the problem set. However, once you learn a topic, you can add any number of problems from this topic to the problem set.

As a result, you are given an array B of C elements, where B_i is the time needed to learn the i^{th} topic. Your task is to print the minimum needed time to prepare this problem set.

Input

The first line contains a single integer T $(1 \leq T \leq 10)$ denoting the number of test cases.

The first line of each test case contains three integers R, N, and C $(1 \leq R \leq N \leq 10^5)$ $(1 \leq C \leq 16)$ denoting the number of needed problems, the number of problems you have and the number of different topics, respectively.

The next N lines contain the description of the problems. The i^{th} line contains two integers S_i and A_i $(1 \leq S_i \leq C)$ $(1 \leq A_i \leq 10^5)$ denoting the type of the problem and the time needed to prepare its test cases.

The next line contains C integers representing B $(1 \leq B_i \leq 10^5)$ denoting the time needed to learn each topic.

Output

For each test case print a single line containing a single integer denoting the minimum time needed to prepare the problem set.

Example

mracpc.in	standard output
1 3 6 5 1 5 2 5 3 4 4 9 1 4 5 1 5 6 7 6 1	16

Problem G. Compression

Input file:	`compression.in`
Output file:	`standard output`
Balloon Color:	`Blue`

El-Khadrawy was trying to come up with a way to reduce the size of the contest files. Therefore, El-Khadrawy invented a new version of ZIP compression that reduces a file of size N bytes to a file of size $floor(N/5)$ bytes.

You are given a file that was compressed using El-Khadrawy's version of the ZIP algorithm. It has a size of X bytes. What is the largest possible size of this file before the compression?

Input

The first line consists of a single integer T, the number of test cases.

Each test case consists of one line containing a single integer X ($1 \leq X \leq 1000$), the size of the file after compression.

Output

For each test case, print a single line containing a single integer, the maximum possible size of the file before the compression.

Example

compression.in	standard output
3 1 2 3	9 14 19

Note

$floor(x)$ means the integer value of x rounded down.

Problem H. Brute Force

Input file:	gcd.in
Output file:	standard output
Balloon Color:	Gold

El-Khadrawy is tired of writing hard problems so he decided to write a problem that has a brute force solution. He came up with the following problem. Given an array a of length n and an array b of length k.

You have to process q queries of two types:

- 1 x y z Print the number of elements in the array a in the range $[x, y]$ which are divisible by b_z.
- 2 x y Change each element in the array a in the range $[x, y]$ to $GCD(a_x, a_{x+1}, \ldots a_y)$.

Input

The first line of the input contains a single integer number T - the number of test cases.

The first line of each test case contains two integer numbers n and k ($1 \leq n \leq 10^5$) ($1 \leq k \leq 50$) denoting the size of the arrays a and b respectively. It's guaranteed that the sum of n over all test cases doesn't exceed 2×10^5.

The next line contains n space-separated integers a_i ($1 \leq a_i \leq 10^9$) denoting the values of the array a.

The next line contains k space-separated integers b_i ($1 \leq b_i \leq 10^9$) denoting the values of the array b.

The next line contains a single integer q ($1 \leq q \leq 10^5$), denoting the number of queries. It's guaranteed that the sum of q over all test cases doesn't exceed 2×10^5.

Then q lines. Each line contains one type of query as described in the statements ($1 \leq x \leq y \leq n$) ($1 \leq z \leq k$).

Output

For each query of the first type print a single line containing a single integer number - the answer to that query.

Example

gcd.in	standard output
1 10 5 3 6 2 4 6 9 24 1 2 3 1 2 3 4 5 5 1 1 5 2 1 1 5 3 1 6 10 4 2 2 3 1 1 3 3	4 3 1 1

Note

After the 4^{th} query, the array will be: $[3, 2, 2, 4, 6, 9, 24, 1, 2, 3]$

Problem I. Cylinders

Input file:	`cylinders.in`
Output file:	`standard output`
Balloon Color:	`Green`

El-Khadrawy has N cylinders, where the i^{th} cylinder has a height H_i and a radius R_i. All of these cylinders are in the backyard of his house.

Each of these cylinders has a water pipe connected to it. El-Khadrawy knows that all these pipes are exactly similar, and they pass the exact same amount of water. Each pipe passes 1 cubic meter of water per minute.

Since it's taking a long time for the cylinders to be filled, El-Khadrawy decided to invent a game and play it with you. Each time, El-Khadrawy will give a query W_j indicating the number of cubic meters of water he needs. So, your job is to tell El-Khadrawy the minimum number of minutes that need to pass before his cylinders contain exactly W_j cubic meters of water. Note that the result can be a floating-point number.

In addition, note that when a cylinder becomes completely filled with water, it doesn't store water anymore. In other words, if a cylinder becomes full, then water keeps spilling to the floor. Of course, the amount of spilled water cannot be counted into the needed amount by El-Khadrawy.

Input

The first line contains a single integer T, denoting the number of test cases.

The first line of each test case contains an integer N ($1 \leq N \leq 10^5$) denoting the number of cylinders.

N lines follow. Each line describes a cylinder with two space-separated integers H_i and R_i ($1 \leq H_i, R_i \leq 10^3$) denoting the height and radius of the i^{th} cylinder.

The next line contains a single integer Q ($1 \leq Q \leq 10^5$) denoting the number of queries.

Q lines follow. Each line contains a single integer number W_j ($0 \leq W_j \leq 10^{18}$) describing the needed amount of water.

Output

For each query print a single line containing a single floating-point number denoting the minimum number of minutes to have W_j cubic meters of water to be filled. If the amount of water cannot be contained inside the cylinders, then print *IMPOSSIBLE*.

Since the answer is a floating-point number, print it rounded to exactly 2 decimal points.

Example

cylinders.in	standard output
1 5 1 1 1 1 2 2 1 1 2 2 4 5 50 15 200	1.00 20.29 3.00 IMPOSSIBLE

Problem J. Functions Algorithm

Input file:	`functions.in`
Output file:	`standard output`
Balloon Color:	`Silver`

You are given N functions. Each function has the following format:

$$F_i(x) = A_i \times x + B_i$$

In addition, you are given multiple queries. Each query can have one of the following types:

- 1 L R X: In this query, you are asked to print the answer produced by the following algorithm:

  $ans \leftarrow x$

  for$(i \in [L, R])$

  $\quad ans \leftarrow F_i(ans)$

  print(ans)

- 2 id A B: In this query, you are asked to update the constants of the i^{th} function to become $F_{id}(x) = A \times x + B$, where A and B are the newly given ones.

Since the answer for the first query can be very large, print it modulo $10^9 + 7$.

Input

The first line contains a single integer T indicating the number of test cases.

Each test case starts with a line containing N and Q $(1 \leq N, Q \leq 10^5)$ indicating the number of functions and the number of queries, respectively.

The next N lines describe the functions. The i^{th} line contains A_i and B_i $(1 \leq A_i, B_i \leq 10^5)$ indicating the constants of the i^{th} function.

The next Q lines describe the queries. Each query has one of the forms:

1 L R X $(1 \leq L \leq R \leq N)$ $(1 \leq X \leq 10^5)$

2 id A B $(1 \leq id \leq N)$ $(1 \leq A, B \leq 10^5)$

Both of them are illustrated inside the statements.

Output

For each query of the first type print a single line containing a single integer - the answer to the query modulo $10^9 + 7$.

Example

functions.in	standard output
1 4 3 1 2 3 4 5 6 7 8 1 1 4 5 2 3 10 12 1 2 4 20	925 4572