Contest Problems

Philadelphia Classic, Fall 2023

University of Pennsylvania

Designed By: Ishaan Lal, Ethan Chee, Grace Lee, Aaron Mei, Nick Eiffert, Kevin Liu, Kyle Xiong, Hemosoo Woo, Michael Gao

Clarifications Doc:

https://tinyurl.com/pcl23FAclarifications

Classic Competition:

https://tinyurl.com/pcl23FAclassic

Advanced Competition:

https://tinyurl.com/pcl23FAadvanced

Rules and Information

This document includes 12 problems.

Classic teams do problems 1-8; Advanced teams do problems 5-12.

Any team which submits a correct solution for any of problems 1-4 will be assumed to be a Classic team. **If you are not a Classic team, please skip problems 1-4.**

Problems 1-4 are easier than problems 5-8, which are easier than problems 9-12.

You may use the Internet **only** for submitting your solutions, reading Javadocs or Python documentation, and referring to any documents we link you to. You may not use the Internet for things like StackOverflow, Google, AI assistance or outside communication. Please do not use ChatGPT.

You are responsible for handling Input and Output on your own.

You will receive 1 point per correct submission. In the case of a tie, the 'total penalty' will be used as a tiebreaker, which is defined to be the sum of time elapsed from the beginning of the contest to the first accepted submission for each problem, plus a 20 minute penalty for each previously rejected run for that problem. Wrong Answer, Time Limit Exceeded, and Runtime Errors all contribute towards the penalty. Unsolved problems do not contribute towards the penalty. There is no partial credit.

Some problems use Java's "long" type; if you are unfamiliar with them, they're like an "int", but with a (much) bigger upper bound, and you have to add "L" to the end of an explicit value assignment:

```
long myLong = 1000000000000L;
```

Otherwise, the "long" type functions just like the "int" type.

Problem A. Prestigious PClassic Preamble

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

"The King has a message for all knights and dames of Pennaloria!"

All the knights at the mess hall turn to watch. The King is holding a new competition to find the best knight of all the lands.

Every knight in the castle has signed up for the competition, but the main contenders are Richard the Rich, Gumbo the Giant, and Ethan the Engineer. Richard the Rich is from the wealthy city of Venice and owns a vast trading empire. Gumbo the Giant is a 10-foot-tall giant hailing from the North from a mysterious frigid kingdom called Adanac. Ethan the Engineer was born in Old Jersey and uses his craftsmanship to design incredible weapons.

All you have is a raggedy old book that your aunt left behind. After careful observation, you notice that some of the words in the book are special. For this problem, you must determine if a word is special. Given an uppercase string s, return YES if the first character of s is one of the characters that appear in PCLASSIC.

Input

The first line contains a string s of uppercase letters ($1 \leq |s| \leq 10^5$).

Output

Output YES if the first character of s is one of the characters that appear in PCLASSIC, else output NO.

You can output the answer in any case (upper or lower). For example, the strings "`yEs`", "`yes`", "`Yes`", and "`YES`" will be recognized as positive responses.

Examples

standard input	standard output
PANTALONES	YES
IGUANA	YES
NINCOMPOOP	NO

Problem B. The Strongest Squire

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

After deciphering the book, the book is a spellbook!

You'll have to learn on the fly because the first challenge is starting. The first competition to determine if you are even eligible for the knight competition is an arm-wrestling competition. You have a base strength of n, and your arm-wrestling prowess is determined by the number of digits in your strength.

Gumbo the Giant goes first, and he quickly throws his opponent to the floor. Ethan the Engineer pulls out a mechanical arm. Is that from STARK industries? Either way, he wins his arm-wrestling competition handily. While the King isn't looking, Richard the Rich hands over a few gold coins to his opponent. He doesn't want to get his gloves dirty. Now it's your turn. After years of eating moldy bread and bits of cabbage, you don't stand a chance against the sour-faced knight sitting across from you.

Page 2 of your spellbook explains a simple strength charm. The strength charm gives you k more strength. You want to know if using the charm would increase your arm-wrestling prowess. Return YES if $n + k$ has more digits than n and NO otherwise.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n and k ($1 \le n, k \le 10^9$) — your base strength and the strength gained by the charm.

Output

For each test case, output YES if $n + k$ has more digits than n, else output NO.

You can output the answer in any case (upper or lower). For example, the strings "`yEs`", "`yes`", "`Yes`", and "`YES`" will be recognized as positive responses.

Examples

standard input	standard output
1 30 78	YES
1 2 5	NO
5 98 1 98 2 98 3 1 10 160 121	NO YES YES YES NO

Problem C. Makin' Magic Money

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Richard the Rich passes by you in his shiny platinum armor. "Look at this peasant!" he cackles.

Unfortunately, he's right. Your leather tunic and wooden sword aren't going to cut it! The blacksmith is right down the cobblestone road, but you only have m coins in your pocket.

Page 3 of your spellbook has a section containing a limited amount of alchemy spells to increase the amount of coins you have in various ways. It contains p multiplicative spells, where the i-th multiplicative spell can multiply the number of coins you currently have by some amount a_i, and q additive spells, where the i-th additive spell can add b_i to the number of coins you currently have.

You can cast each spell at most once, and you want to cast the spells in the order that maximizes the number of coins you end up with. After casting your spells in the optimal order, how many coins do you have?

Input

The first line contains a single integer m ($1 \le m \le 100$) — the number of coins you have in your pocket to start.

The second line contains a single integer p ($1 \le p \le 5$) — the number of multiplicative spells you have.

The third line contains p integers $a_1, a_2, \ldots, a_p$ ($1 \le a_i \le 4$) — the multiplicative spells.

The fourth line contains a single integer q ($1 \le q \le 5$) — the number of additive spells you have.

The fifth line contains q integers $b_1, b_2, \ldots, b_q$ ($1 \le b_i \le 1000$) — the additive spells.

Output

Output a single integer, the maximum number of coins you can end with after using the spells.

Examples

standard input	standard output
100 1 4 1 10	440
10 4 2 2 2 2 1 100	1760
10 1 2 4 10 10 20 500	1100

Problem D. Dicing up Dummies

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

The next challenge tests your longsword slashing skills. A rectangular dummy stands in front of you. It's a race to see who can cut through the dummy the fastest!

Page 5 of the spellbook allows you to fine-tune your motor skills, allowing you to make precise incisions into the dummy.

The rectangle dummy is bounded by a y-axis ($x = 0$), an x-axis ($y = 0$), and 2 perpendicular lines, $x = a$ (the other vertical side) and $y = b$ (the other horizontal side). You slash n vertical and m horizontal lines which divide the rectangle dummy into smaller sub-rectangles. The slashes are only made at integer numbered distances. You are experienced, so each slash is guaranteed to be within or on the rectangle. Find the area of the largest rectangle formed after the slashes (these rectangles cannot have any internal dividing lines).

Hurry up! Gumbo the Giant is already punching through his dummy, and Richard the Rich is trying to buy one of those chainsaws that Ethan the Engineer made (is that even allowed?)

Input

The first line contains two integers a and b ($1 \le a, b \le 10^3$) — the positions of vertical and horizontal sides.

The second line contains a single integer n ($1 \le n \le a$) — the number of vertical slashes.

The third line contains n integers $x_1, x_2, \ldots, x_n$ ($0 \le x_i \le a$) — the vertical slashes, where the i-th slash is the vertical line $x = x_i$, in increasing order. It is guaranteed that all vertical slashes are unique.

The fourth line contains a single integer m ($1 \le m \le b$) — the number of horizontal slashes.

The fifth line contains m integers $y_1, y_2, \ldots, y_m$ ($0 \le y_i \le b$) — the horizontal slashes, where the i-th slash is the horizontal line $y = y_i$, in increasing order. It is guaranteed that all horizontal slashes are unique.

Output

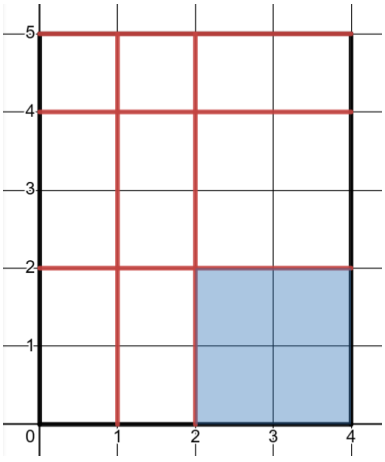
Output a single integer, the area of the largest rectangle formed after the slashes.

Examples

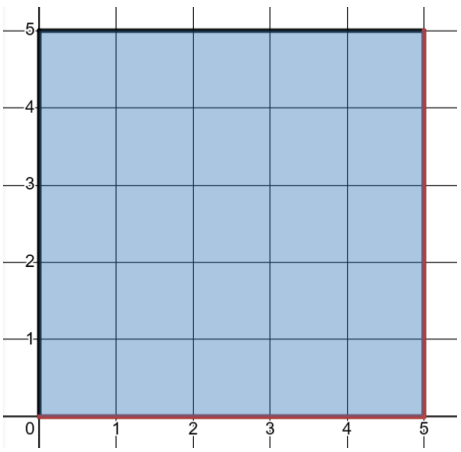
standard input	standard output
4 6 2 1 2 3 2 4 5	4
5 5 1 5 1 0	25
5 7 1 4 1 0	28

Note

In the first test case, the rectangle is 4 wide and 6 high. The slashes divide it into 4 (four) 1×1, 6 (six) 1×2, and 2 (two) 2×2 sub-rectangles. The largest area is shared by the 2 2×2 sub-rectangles, so the largest sub-rectangle has area $2 \times 2 = 4$.

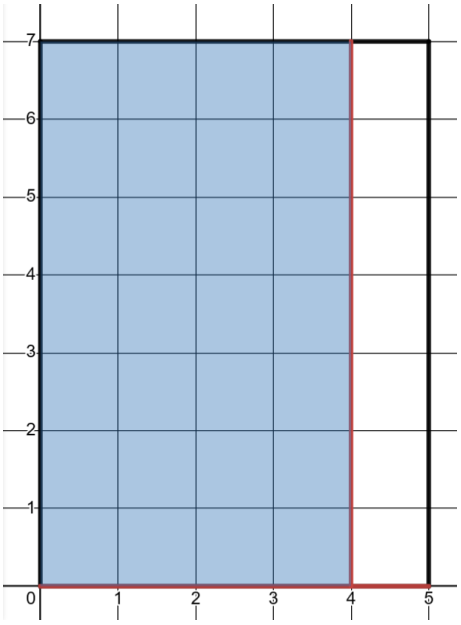


In the second test case, the rectangle is 5 wide and 5 high, and all the slashes ($x = 5$, $y = 0$) lie on the rectangle itself, so the sub-rectangle of the largest area is $5 \cdot 5 = 25$.



In the third test case, the rectangle is 5 wide and 7 high with slashes $x = 4$ and $y = 0$. The largest

sub-rectangle is bounded on the right by $x = 4$, and it has area $4 \cdot 7 = 28$.



Story Summary

It's the year 1466 and you are a lowly squire living in the kingdom of Pennaloria. Your distant aunt has recently passed away, and she's left behind her magic spellbook!

The King is holding a new competition to find the best knight of all the lands: the Ultimate Knight. The main contenders are Richard the Rich, Gumbo the Giant, and Ethan the Engineer. Richard the Rich is from the wealthy city of Venice and owns a vast trading empire. Gumbo the Giant is a 10-foot-tall giant hailing from the North from a mysterious frigid kingdom called Adanac. Ethan the Engineer was born in Old Jersey and uses his craftsmanship to design incredible weapons.

No time to waste! You're also in the tournament, competing amongst the best. Use your spells to become the Ultimate Knight!

Problem E. Ring Catching

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Giddy-up! No knight is complete without extensive horse-riding skills. Ring-tilting is a common method used by knights to build the skill of targeting a lance, and involves using a spear to catch rings hanging from a post. Richard the Rich laughs as he rides past you on his one-million-gold-coin stallion.

You don't even own a saddle! Luckily, page 7 of the spellbook describes methods for talking to animals. Let's ride!

You are positioned at one end of a field of posts with rings. The field is an array of n posts, where the value in each cell represents the number of rings at that post. You start on your horse at the leftmost post. You have k turns and at each turn, you can either move left, right, or stay at the same post. Then you will automatically collect all the rings at the post you end your turn at. Rings are replenished after each turn. Find the maximum number of rings you can catch.

Input

The first line contains two integers n and k ($1 \le n, k \le 2 \cdot 10^5$) — the number of posts and turns.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^3$) — the number of rings at each post.

Output

Output a single integer, the maximum number of rings you can catch after k turns.

Examples

standard input	standard output
2 1 4 2	4
5 4 1 2 3 4 5	14
5 5 1 2 10 9 9	42

Note

In the first test case, you have one turn, so you can either choose to stay at the leftmost post and collect 4 rings or move to the right and collect 2 rings. Therefore, the maximum number of rings you can collect is 4.

In the second test case, the sequence of moves that gives you the most rings is right to collect 2 rings, right to collect 3 rings, right to collect 4 rings, then right to collect 5 rings.

In the third test case, the sequence of moves that gives you the most rings is moving right to collect 2 rings, right to collect 10 rings, and then staying to collect 10 rings for the remaining turns.

Problem F. Traversing Through Trees

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

After the longsword challenge, only four contestants remain: Richard the Rich, Gumbo the Giant, Ethan the Engineer, and you.

The next challenge takes place in a clearing in the middle of the treacherous forests. The four of you need to navigate through the forest to the clearing. Gumbo the Giant is already stampeding through the wild, Ethan the Engineer is building a satellite GPS device, and Richard the Rich is paying a group of peasants to carry him through the forest.

Page 11 of the spellbook has a magical map of the forest. Unfortunately, reading the map requires using magical numbers. The map says:

Numbers in the magical format are similar to normal decimal numbers with one exception. Instead of every digit ranging from 0 to 9, the numbers each digit can range from are dependent on how many digits are to the right of the digit. For example, the first digit, the digit furthest to the right, has values ranging from 0 to 1. The second digit has values ranging from 0 to 2. If a digit can have more than 10 values then its values will be represented by uppercase letters, starting from A going to Z in alphabetical order. For example, the digit that has 12 possible values will have values ranging from 0 to B. You do not have to worry about running out of letters to use as the numbers will not grow that big.

For example, the numbers 1 to 6 are represented as 1, 10, 11, 20, 21, and 100.

Given a normal decimal number n, return the equivalent magical number. If you can do this, you can navigate the treacherous forest safely!

Input

The first line contains a single integer n ($1 \leq n \leq 10^9$).

Output

Output a single string, the number n in magical format.

Examples

standard input	standard output
21	311
84	3200
315	23011
39916799	A987654321

Problem G. Arbor Annihilation Archery

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Ethan the Engineer never made it to the clearing in the forest. Legend says he got trapped into organizing a coding competition...

Richard the Rich, Gumbo the Giant, and you face a row of n trees of varying heights h_i, all in a straight line. You stand at the left end, facing right toward the line of trees. With your archery skills, you can shoot a tree as long as all trees to the left of it are strictly shorter than it.

Page 13 of the spellbook gives you a spell to remove a tree. Choose one tree to remove that maximizes the number of trees you can shoot.

You must remove one tree. In the case of a tie, you should remove the leftmost tree.

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of trees.

The second line contains n integers $h_1, h_2, \ldots, h_n$ ($1 \le h_i \le 10^9$) — the heights of the trees from left to right.

Output

Output a single integer, the position of the tree to remove.

Examples

standard input	standard output
4 1 5 2 3	2
7 1 8 3 2 11 9 10	5
7 1 2 5 3 4 100 101	3

Note

In the first test case, removing the second tree (position 2) allows you to hit tree 3 and tree 4

In the second test case, removing the tree at position 5 of height 11 will allow 4 trees to be hit.

In the third test case, removing the tree at position 3 of height 5 will allow 6 trees to be hit.

Problem H. Tricky Typing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Richard the Rich is disqualified in the archery round. Apparently, he tried to bribe the King for the Ultimate Knight role. "I'll make you pay for this!" he says as he heads back into the forest. The King leads you and Gumbo the Giant to the grand palace doors. The only problem is that there is a cobblestone wall blocking the entrance to the palace. "The first person to break the wall and enter the palace will be knighted as the Ultimate Knight!" says the King. Gumbo the giant is already pummeling the cobblestone wall with his bare hands. It looks like he's going to get through in a few minutes, better think fast!

On page 17 of the book, there is a charm to break a cobblestone wall that uses precise pressure points. The wall is made up of a 5 by 5 square grid labeled with 25 unique characters of the English alphabet (missing 1) in a specified order.

Your left hand and right hand start at letters ℓ and r respectively. In order to cast the spell, you need to move your hands and press the bricks to spell a given input string s. You are guaranteed that the missing letter does not appear in s.

Each of your hands can move to an adjacent brick (diagonal moves are allowed). You can only press a brick that either your left or right hand has reached. Pressing a brick takes 1 second and moving one hand to an adjacent brick takes 1 second. You can only perform one action at a time, whether it be moving one of your hands or pressing a brick. Compute the minimum time it takes to type a given input string s and successfully cast the spell. It is guaranteed that all characters in s can be found on the 5 by 5 grid of characters.

Input

The first 5 lines contain 5 lowercase letters each. It is guaranteed that each of the 25 characters is unique.

The next line contains two characters ℓ and r — the starting position of your left and right hand.

The next line contains a single string s consisting of lowercase letters ($1 \leq |s| \leq 10^4$).

Output

Output a single integer, the minimum time in seconds it takes to type s.

Examples

standard input	standard output
mixzd qwgsc bjynv kflot aephr d b xqm	7
abcde fghij klmno pqrst uvwxy e u aytp	14
upmbi yhogv cnfas qkwzd telxr a a pclassic	19

Note

In the first test case, we can enter the string as follows:

We first move the left hand left 2 to press x in $2 + 1 = 3$ seconds. We then move the right hand up 1 to press q in $1 + 1 = 2$ seconds. We then move the right hand from q to m and press in $1 + 1 = 2$ seconds. This takes $3 + 2 + 2 = 7$ seconds overall.

Problem I. Medieval Massacre

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

You break the cobblestone wall and enter the palace. You are the Ultimate Knight! Gumbo the Giant follows a few minutes after and congratulates you. The King is about to knight you as the Ultimate Knight when a messenger runs in. "It's Richard the Rich!" the messenger screams. "He's brought his Venetian army!" Apparently, Richard the Rich feels he was cheated out of the title of Ultimate Knight, and he intends to take the prize by force.

A battle ensues between the King's army and Richard the Rich's army. Both armies have the same number of n knights. The knights are trained exactly the same way, making them have the same health h and attack k. We denote the King's knights as $a_1, ..., a_n$ and Rich's knights $b_1, ..., b_n$. Each second, all knights that are still alive will attack a knight of the opposing nation, making them lose k health. When a knight reaches non-positive health, they are considered dead, and they are unable to attack or be attacked. All knights attack simultaneously, that is a knight may attack on the same second it dies.

Richard the Rich has a specific, predetermined order to attack the King's knights. That is, for every knight b_i, at any given second, b_i will attack some knight a_j. If a_j is dead, knight b_i does nothing. However, the King allows you to direct his knights freely. Hence, for every knight a_i, as long as a_i is not dead, a_i can attack some knight b_j where b_j can change every second. You must help the King find a strategy to defeat Richard the Rich's knights. If you direct the knights optimally, find the minimum number of seconds it takes for your knights to eliminate the knights of Richard!

Input

The first line contains three integers n, h, and k ($1 \le n \le 9$, $1 \le h, k \le 10^6$) — the number of knights on each side, and the health and attack of a knight.

The second line contains n integers $t_1, t_2, \ldots, t_n$ ($1 \le t_i \le n$) — the assignment of Richard's knights to attack the King's knights, where b_i attacks a_{t_i}.

Output

Output a single integer, the minimum number of seconds it takes for your knights to defeat Richard's knights.

Examples

standard input	standard output
4 4 1 4 3 1 1	4
3 50 20 1 2 3	3
9 3 1 1 1 1 2 2 2 3 3 3	4

Note

In the first test case, one optimal way to direct the knights is as follows:

On the first turn, all knights target b_4. On the second turn, all knights target b_3. On the third turn, two knights target b_1, and two knights target b_2. On the fourth turn, two knights target b_1, and two knights target b_2.

In the second test case, one optimal way to direct the knights is as follows:

On each turn, each knight targets its mirror. After 3 turns, all knights are dead.

Problem J. Gumbo the Gallant Giant

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

The King is wounded badly in Richard the Rich's attack! He has put you in charge of the army! If you can conquer Italy all the way to Venice and force Richard the Rich to surrender, then you will be knighted as the Ultimate Knight.

There are n cities that need to be defeated in Italy on the path to Venice. Luckily, Gumbo the Giant has agreed to help you. He has maximum health m and can destroy an entire city by himself. At the i-th city, Gumbo the Giant takes a_i damage. When Gumbo the Giant's health hits 0, he dies. It looks like there are too many cities for Gumbo the Giant to make it through.

Page 19 of the spellbook describes a heal spell. If you place a heal spell at city i, then for each city in the range $[i, i+k-1]$, Gumbo heals h amount of health. More formally, if Gumbo arrives at the i-th city with health x, if there is an active heal spell, he leaves the city with health $min(m, x - a_i + h)$ (he cannot heal more than his maximum health). You can assume that the health healed and the damage taken occur simultaneously.

Note that heal spells do not overlap. Output the minimum number of heal spells you need to use to save Gumbo. Also, output the positions you place the heal spells in order of city. If it is impossible to save Gumbo, output -1.

Input

The first line contains four integers, n, m, h, and k ($1 \le k \le n \le 10^6$, $1 \le h, m \le 10^9$) — the number of cities, Gumbo's maximum health, heal spell's heal rate, and length of a heal spell.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$) — the damage taken at each city.

Output

If it is impossible to save Gumbo, output -1. Otherwise, output a single integer L followed by a list of L distinct positions in increasing order, on the same line and separated by spaces, where L represents the minimum number of heal spells required and the following L positions represent placements of the heal spells.

Examples

standard input	standard output
4 1 1 1 1 1 1 1	4 1 2 3 4
5 4 5 1 4 3 0 0 1	2 1 2
10 10 5 3 5 10 3 5 4 5 10 0 2 3	3 1 4 7
5 6 4 2 3 9 2 7 3	-1
5 6 1 1 1 1 1 1 1	0

Note

In the first test case, we must place a heal spell at each position for the giant to survive.

In the second test case, we can place a heal spell at position 1 and a heal spell at position 2. It can be shown that this is the minimum amount needed.

In the fourth test case, no matter how we place the heal spells, the giant will still die.

Problem K. Grand Canal Dynamics

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Gumbo the Giant has led the King's army all the way to Venice. You thank him for his heroic service.

Now it's time to make Richard the Rich pay! Venice is a city full of canals which you were not expecting. How is your army going to navigate through all those canals?

Ethan the Engineer has returned! He has agreed to build bridges over the canals. Bridges can be conjured by reciting a "magic sequence". A magic sequence a can be constructed as follows:

Start with $a_1 = 1$ and $a_2 = 1$.

Do the following zero or more times: Let ℓ be the current length of the sequence. Choose some positive integer k_ℓ and add $k_\ell \cdot a_\ell + a_{\ell-1}$ to the end of the sequence.

What is the number of magic sequences where all of the terms are bounded by n? Output your answer modulo $10^9 + 7$.

Input

The first line of the input contains one integer t ($1 \le t \le 10^4$) — the number of test cases.

Each test case contains one integer n ($1 \le n \le 2 \cdot 10^5$) — the bound for the magic sequences.

Output

For each test case, output the number of magic sequences that are bounded by n modulo $10^9 + 7$.

Example

standard input	standard output
4 4 10 300 199234	6 32 27398 65622424

Note

In the first test, there are 6 magic sequences whose terms are all bounded by 4: $[1, 1]$, $[1, 1, 2]$, $[1, 1, 3]$, $[1, 1, 4]$, $[1, 1, 2, 3]$, and $[1, 1, 3, 4]$.

Problem L. Tactical Tournament Troop Training

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Thanks to Ethan the Engineer's bridges, your army has finally reached the Venetian Palace where Richard the Rich lives. A brawl has broken out!

There are n knights that are fighting. The battle occurs in rounds, where the 1st and 2nd knights face each other, the 3rd and 4th face each other, and so on. After that, the winner of the first match competes with the winner of the second match, etc. The tournament ends when there is only one knight left. Basically, this is a single-elimination tournament.

Each knight is either part of the King's army or Richard the Rich's army, where the i-th knight is part of army t_i, with $t_i = 0$ representing Richard's army and $t_i = 1$ representing the King's army. The score of an army is the total number of matches won by knights in that army during the tournament.

The winner of each match is determined by the knight's power levels, where the i-th knight has power p_i. The knight with the higher power level always wins the match. If two knights have the same power level, then if they are part of the same army, a knight will advance arbitrarily. If two knights have the same power level and are part of different armies, then the knight who is part of the King's army will advance.

But before the battle can start, you have q spells to cast. Page 23 of the spellbook describes a collection of spells that boost the morale of knights. When using the j-th spell, you boost the power level of all the King's knights in a segment from l_j to r_j by b_j, but not any of Richard's knights.

After casting the j-th spell, find the score of the King's army at the end of the battle if the first j spells were cast.

Input

The first line contains two integers n and q ($1 \le n, q \le 10^5$) — the number of knights and the number of spells. It is guaranteed that n is a power of 2.

The second line contains n integers $t_1, t_2, \ldots, t_n$ ($t_i \in \{0, 1\}$) — the team of each knight.

The third line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le 10^9$) — the power level of each knight before spells.

The j-th of the next q lines contains three integers l_j, r_j, and b_j ($1 \le l_j \le r_j \le n$, $1 \le b_j \le 10^9$) — the left and right bounds of the segment and the power level boost for the j-th spell.

Output

Output q integers, one per line, where the j-th integer is the score of the King's army at the end if the first j spells were cast.

Examples

standard input	standard output
4 3 0 1 1 0 6 4 3 9 1 3 1 1 2 1 3 4 6	0 1 3
8 5 1 0 1 0 1 0 0 1 2 10 4 13 1 20 23 11 5 8 11 2 7 8 1 7 1 1 8 3 1 2 5	0 1 3 6 7

Note

In the first test case, the powers and scores are updated as follows:

After the first boost, the powers of knights 2 and 3 are boosted, so the powers are 6 5 4 9. In the first round, knight 1 wins ($6 > 5$), and knight 4 wins ($9 > 4$). In the second round knight 4 wins ($9 > 6$). Since knight 2 and knight 3 never win, the score of the King's knights is 0.

After the second boost, the powers of knight 2 is boosted, so the powers are 6 6 4 9. In the first round, knight 2 wins ($6 \geq 6$), and knight 4 wins ($9 > 4$). In the second round knight 4 wins ($9 > 6$). Since knight 2 wins 1 match and knight 3 wins no matches, the score of the King's knights is 1.

After the third boost, the powers of knight 3 is boosted, so the powers are 6 6 10 9. In the first round, knight 2 wins ($6 \geq 6$), and knight 3 wins ($10 \geq 9$). In the second round knight 3 wins ($10 > 6$). Since knight 2 wins 1 match and knight 3 wins 2 matches, the score of the King's knights is 3.

Epilogue

The King is delighted to hear that you've captured Richard the Rich! Richard the Rich is being dragged to the dungeons where he will share a cell with SBF and Elizabeth Holmes.

Now, for the festivities! The King begins by declaring the Ultimate Knight title to...

Gumbo the Giant and Ethan the Engineer!

The crowd in the palace cheers. Gumbo the Giant and Ethan the Engineer kneel as the King formally knights them.

Then the King steps down from his throne and hands the Royal Crown to you!

"I've come to realize this King business is pretty dangerous," he says. "I'm naming you the new King of Pennaloria!" Then he puts on a pair of Crocs and waltzes into retirement.

Congratulations! You're the King of Pennaloria!