mBIT Varsity

November 2, 2019

These problems are roughly sorted in order of difficulty. However, we suggest you look at and think about as many problems as you can in the time provided. Good luck and happy coding!

Contents

Thanks to Evan Chen for letting us use his style file

The memory limit for all problems is the standard 256 MB.

Time Limits (s)			
Problem	C++	Java	Python
Baking Pan	1	2	2
Frosting Patterns	1	1	1
Mountain Mileage	1	1	2
Milky Way	1	1	2
Coffee Swapping	1	2	2.5
Hot Cake	2	2	2
Cupcake Distribution	1	1	3
Secret Base	2	2	5
Contradictory Canelé	1	1	1
Cake Cutting	1	2	3
Lost Child	2	2	4
Outbreak	2	3	6

§1 Baking Pan

George wants to bake N cookies, but he lost his baking pan. He wants to get a new one with the smallest area possible. Also, he wants to bake these cookies in a certain aesthetic design. Since George is a really good baker, all of his cookies come out circular. He has planned out his design on a coordinate plane with integer coordinates for the center of each cookie and an integer radius for each cookie. Given this information and the fact that the baking pan's edges must be parallel to the x and y-axes of his design, can you determine the smallest area of a baking pan that contains all of the cookies George wants to bake? Note that the cookies may overlap.

Input Format:

The first line contains the number of cookies George wants to bake, N. ($1 \leq N \leq 10^5$)

The next N lines each contain the center x coordinate, center y coordinate, and radius for a cookie. ($-10^7 \leq x_i, y_i \leq 10^7$ and $1 \leq r_i \leq 10^7$)

Output Format:

Output the area of the smallest possible baking pan George can use. (Beware of integer overflow error if you are using C++ or Java)

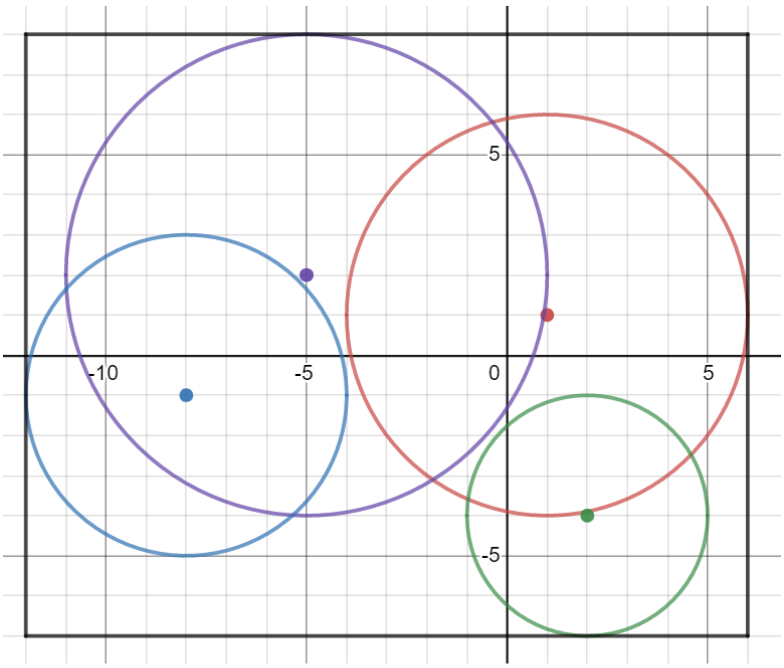
Sample Input:

```
4
1 1 5
2 -4 3
-5 2 6
-8 -1 4
```

Sample Output:

```
270
```

The smallest possible baking pan has dimensions 18×15.



§2 Frosting Patterns

Aunt Amy is decorating a cake for her nephew's birthday. She plans on making a pattern with the colors of the frosting dots. Each color is represented with an uppercase letter. For example, one pattern Amy could choose is:

ABBDCABBDCABBDCA...

Here, the block of colors ABBDC is repeated over and over.

Amy has started her pattern, but needs help finishing it. She has already placed down N colors. Can you tell her what the next M colors in the pattern should be? It is guaranteed that the repeated block of colors has already occurred at least twice.

Input Format:

The first line contains N and M. $(1 \leq N, M \leq 1000)$

The next line contains N uppercase letters, representing the colors Amy has already placed.

Output Format:

Output a single line with M uppercase letters representing the next M colors in the pattern

Sample Input:

```
19 4
MBITTIMEMBITTIMEMBI
```

Sample Output:

```
TTIM
```

The repeated sequence of colors in this case is MBITTIME.

§3 Mountain Mileage

There are N friends who live at various heights on the side of a mountain. One day, the friends decide to have a cookie swap party at one of their houses. They want to hold the party at the house that minimizes the total gallons of gas they need for everyone to drive there.

The gas mileage of their cars is different depending on whether they drive uphill or downhill. It takes $A(y-x)$ gallons of gas to drive from a house at height x to a house at height y, if $x < y$. It takes $B(x-y)$ gallons to drive from a height of x to a height of y, if $x \geq y$. The friends who live at the same height as the party house do not drive.

If the friends choose the optimal house, how much gas will they need altogether? Be sure to account for integer overflow errors if you are using Java or C++.

Input Format:

The first line will contain N, A, and B. $(1 \leq N, A, B \leq 10^5)$
The next line will contain N integers, representing the heights of the houses on the mountain. The heights lie between 0 and 10^5, inclusive.

Output Format:

Output a single integer representing the total amount of gas the friends need for everyone to reach the party house.

Sample Input:

```
5 4 3
1 6 7 2 9
```

Sample Output:

```
48
```

When the house at height 6 is chosen, the total amount of gas required is $4((6-1)+(6-2))+3((9-6)+(7-6)) = 48$. This is the optimal choice.

§4 Milky Way

Interplanet Janet has just built the first ever sugar-powered spaceship. There are N planets in her region of the Milky Way numbered $1, 2, \ldots, N$. Janet is currently on planet A and her mission is to reach planet B as soon as possible. The spaceship works by completing "hyperjumps" from planet to planet. Due to complications in quantum electrodynamics and general relativity, there are only certain pairs of planets which Janet's ship can hyperjump between. There are M of these hyperjump possibilities. It takes Janet 1 day to complete each hyperjump.

Janet also has to worry about fuel. Hyperjumping from one planet to another expends a certain amount of fuel units. Fortunately, the planets in this region of the Milky Way are very sweet, and the spaceship is powered by sugar. Janet can spend as many days as she wants along her journey refueling her ship instead of completing hyperjumps. Different planets have different sweetness levels given by $P_1, P_2, \ldots, P_N$. Refueling for a day on planet i will give fill up the spaceship's tank by P_i fuel units. The ship can only hold up to C fuel units at a time, and it cannot hyperjump between two planets if doing so requires more fuel than is currently in the tank.

Please tell Janet the minimum number of days she needs to reach planet B starting from planet A.

Here are some important points:

- Janet may visit and refuel from any planet more than once.
- The spaceship starts the journey with no fuel.
- Janet may choose to refuel multiple days in a row.
- Refueling on planet i increases the fuel by P_i units, but the total fuel is capped at C units.
- Janet cannot refuel for a fraction of a day to fill the tank by a fraction of P_i.
- If Janet can hyperjump from planet i to planet j, she can also hyperjump from planet j to planet i.
- More experienced competitors can think of the planets as nodes of a graph, with undirected weighted edges representing which planets she can hyperjump between.

Input Format:

The first line contains N (the number of planets), M (the number of viable pairs of planets the ship can hyperjump between), and C (the maximum capacity of the tank). ($2 \leq N \leq 100$ and $N - 1 \leq M \leq \binom{N}{2}$ and $1 \leq C \leq 100$)

The second line contains A and B, the starting and ending planets. ($1 \leq A, B \leq N$)

The third line contains N integers: $P_1, P_2, \ldots, P_N$. ($1 \leq P_i \leq C$)

The next M lines each contain three integers. Say line l contains x_l, y_l, and z_l. This means that Janet can hyperjump from planet x_l to planet y_l (or the other way around). This hyperjump requires z_l fuel units to be expended. ($1 \leq x_l, y_l \leq N$ and $1 \leq z_l \leq C$)

Output Format:

Output a single integer representing the minimum number of days Janet needs to reach planet B starting from planet A.

Sample Input:

```
6 7 16
1 5
2 2 4 5 1 8
1 6 1
1 2 2
2 3 5
3 4 1
4 2 3
4 1 4
5 2 15
```

Sample Output:

```
8
```

Janet can complete the journey in 8 days:
Day 1: Refuel at planet 1 (now has 2 fuel units)
Day 2: Hyperjump to planet 6 (spends 1 fuel unit, now has 1 unit)
Day 3: Refuel at planet 6 (now has 9 fuel units)
Day 4: Refuel at planet 6 (now has 16 fuel units, tank is completely full)
Day 5: Hyperjump to planet 1 (spends 1 fuel unit, now has 15 units)
Day 6: Hyperjump to planet 2 (spends 2 fuel unit, now has 13 units)
Day 7: Refuel at planet 2 (now has 15 fuel units)
Day 8: Hyperjump to planet 5 (spends all 15 fuel units)

§5 Coffee Swapping

Ben, the barista, works at a cafe. He is setting up a line of coffee cups as a display for customers. Currently, there are $2N$ filled coffee cups with distinct temperatures lined up in a row. The temperatures of the cups can be listed as $T_1, T_2, \ldots, T_{2N}$. Ben is a perfectionist, and over the years he has developed a formula for determining the *beauty* of the line of cups. His formula for B, the beauty, is:

$$B = (T_1 + T_2) \cdot (T_3 + T_4) \cdot \dots \cdot (T_{2N-1} + T_{2N})$$

Ben wants to make the line of cups as beautiful as possible (he wants to maximize B), but he can't change any of the temperatures themselves. To achieve his goal, he can only rearrange the order of the cups by making a series of swaps where each swap switches the values of T_i and T_j for any i and j. What is the least number of swaps he needs in order to maximize the beauty of the line of cups?

Input Format:

The first line will contain N. $(1 \leq N \leq 10^5)$

The next line will contain $2 \cdot N$ distinct integers representing the temperatures of the cups in the row. $(0 \leq T_i \leq 10^9)$

Output Format;

Output a single number representing the minimum number of swaps Ken needs to achieve the maximal beauty.

Sample Input

```
3
4 1 2 9 3 6
```

Sample Output

```
2
```

One way to maximize beauty is when the cups are in the order $[9, 1, 2, 6, 3, 4]$.
The beauty is $(9 + 1)(2 + 6)(3 + 4) = 560$.
This order can be achieved in two swaps: First swap the 4 and 9, then swap the 4 and 6.

§6 Hot Cake

N people are playing a game called "hot cake". The game is very similar to "hot potato": A hot cake is passed around the group and the player who is holding the hot cake when the music ends is out. The players are numbered $1, 2, \ldots, N$. You will be given a sequence of integers $P_1, P_2, \ldots, P_N$ that represents the way the people pass the hot cake to each other. Whenever person i has the hot cake, he or she will always pass it to person P_i next. Each pass takes 1 second.

Person 1 starts with the cake. After one second, person P_1 will have the cake. After two seconds, person P_{P_1} will have the cake, and so on. Who will have the cake after K seconds, when the music ends?

Input Format:

The first line will contain N and K. ($1 \leq N \leq 10^6$ and $1 \leq K \leq 10^9$)

The next line contains N integers: $P_1, P_2, \ldots, P_N$. ($1 \leq P_i \leq N$)

Output Format:

Output single integer representing the number of the person who will have the cake after K seconds.

Sample Input:

```
5 10
2 3 1 5 4
```

Sample Output:

```
2
```

The cake will pass from person 1 to 2 to 3 and then back to 1, and so on. After 10 seconds, person 2 will end up with the cake.

§7 Cupcake Distribution

Chef Kevin is the leader of a team of chefs. The team must provide cupcakes for N customers standing in a line, numbered $1, 2, \ldots, N$. Each customer has a preference for the sweetness level of their cupcake. The sequence of integers $S_1, S_2, \ldots S_N$ represents their preferences, with customer i preferring a sweetness of level of S_i.

Kevin's job is to partition up the line of customers into groups, so each group is served by a single chef. The chefs are prideful, so no group in Kevin's partitioning may have less than K customers. In other words, Kevin must choose $A_1, A_2, \ldots, A_m$ such that the first A_1 customers are in a group, the next A_2 customers are in a group, and so on. $A_1 + A_2 + \cdots + A_m = N$, and $A_j \geq K$ for $j = 1, 2, \ldots, m$. There are an infinite number of chefs, so Kevin may form as many groups as he wants (as long as each one has at least K people).

The chefs are also lazy, so each one will only bake cupcakes of one sweetness for his or her whole group. If customer i receives a cupcake of sweetness X, he or she will have an unhappiness of $(S_i - X)^2$. Each chef will choose to bake cupcakes of a sweetness level that minimizes the total sum of unhappiness of customers in his or her group. X does not have to be an integer.

For example, say that a chef serves customers $L, L+1, \ldots, R$. The chef will choose X that minimizes $(S_L - X)^2 + (S_{L+1} - X)^2 + \ldots (S_R - X)^2$.

Please determine the sum of unhappiness of all customers if Kevin partitions the line optimally and each chef chooses an optimal sweetness level. Return the floor of the answer.

Input Format:

The first line will contain two integers N and K.$(1 \leq K \leq N \leq 2500)$

The next line will contain N integers representing $S_1, S_2, \ldots, S_N$. $(1 \leq S_i \leq 10^6)$

Output Format:

Output a single integer representing the minimum unhappiness level, truncated to remove any decimal part.

Sample Input

```
9 4
4 9 6 5 11 25 18 20 16
```

Sample Output

```
78
```

In this case, the optimal choice for Kevin is to group the first 5 customers together and the last 4 together. The first chef bakes cupcakes at a sweetness of 7, and the second chef bakes at a sweetness of 19.75. The total unhappiness is: $(4-7)^2 + (9-7)^2 + (6-7)^2 + (5-7)^2 + (11-7)^2 + (25-19.75)^2 + (18-19.75)^2 + (20-19.75)^2 + (16-19.75)^2 = 78.75$ The output is $\lfloor 78.75 \rfloor = 78$.

§8 Secret Base

General Tanya is in charge of safeguarding Candyland from attacks. Candyland consists of N villages labeled $1, 2, \ldots, N$ connected by $N - 1$ paths. It is possible to reach every village from every other village by traveling along these paths, and each path can be traveled on in both directions.

Tanya wants to set up a military base in one of the villages. She needs to make sure her council of K advisors can evacuate the base if it is infiltrated by Candyland's enemy, Veggieland. To do this, she must build K escape routes in the form of licorice tunnels. Each tunnel must start in the village with the military base and can end in any other village. In the event of an attack, each one of her advisors will use their own tunnel. All K advisors must then be able to meet up by traveling along the normal paths between villages. However, they may *not* go through the village with the base in it to get to this meeting point (it is under enemy control). They also may not use the tunnels again, because that would bring them back to the base.

Unfortunately, licorice tunnels are expensive and only possible to build in certain locations. In fact, there are only M tunnel blueprints that Tanya's architect has designed. Each blueprint describes the endpoints and cost of a tunnel that Tanya can build (in chocolate coin currency). Tanya's goal is to choose a base village and K of the M tunnel designs that satisfy the aforementioned conditions and minimizes the cost of construction. If Tanya chooses optimally, how much will this cost? Building the base is free: only the tunnels themselves must be paid for.

Here is a more formal description of Tanya's task. Say she chooses to build her military base at village H. If the K tunnels she chooses to construct connect H with villages $V_1, V_2, \ldots, V_K$, Tanya wants all the villages V_i to be in the same connected component of the graph when village H and all of its edges are removed. Remember that Tanya gets to choose H and the K tunnels she builds. She wants to minimize the sum of costs of the tunnels.

Input Format:

The first line contains three integers N, M, and K. $(1 \leq N, M, K \leq 10^5)$

The next $N - 1$ lines contain the paths in the village. Each line contains two integers, a and b, which represents the two villages the path connects. $(1 \leq a, b \leq N)$

The next M lines describe the blueprints of licorice tunnels. Each line contains three integers, u, v, and w. This represents the blueprint of a tunnel that would connect u and v. It would cost w chocolate coins to build it. $(1 \leq u_i, v_i \leq N$ and $0 \leq w_i \leq 10^9.)$

Output Format:

Output a single integer representing the minimum amount of money (in chocolate coins) Tanya needs build the K tunnels, assuming she made her decisions optimally.

Sample Input:

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

Sample Output:

```
3
```

Since all possible tunnels are connected to village 3, the base must be located at 3. Tanya should build tunnels $(3, 6)$ and $(3, 4)$, because it is possible can reach village 6 from village 4 without going through village 3. This is the only set of tunnels that works, thus our answer is $1 + 2 = 3$.

§9 Contradictory Canelé

When referencing rectangle grids in this problem, the first number is the row and the second is the column.

Chef Erin has been tasked with preparing a rectangular $N \times M$ platter of canelés in her exam to become a Certified Master Chef. The judge can't possibly consume all of the canelés, so he will only eat canelés in an $A \times B$ subrectangle of the platter. Before she starts cooking, she must plan out the sweetness level of every canelé on the platter according to certain requirements.

As a rule, each canelé's sweetness is represented by an integer sweetness level from -10^9 to 10^9. Firstly, she doesn't want to leave the judge with a bitter mouthful. This means that the sum of the sweetness levels of the canelés the judge eats must be nonnegative. However, Erin doesn't know which subrectangle of canelés the judge will eat, so she must make sure every possible $A \times B$ subrectangle of the platter has a nonnegative sweetness sum.

Furthermore, since bitter canelés are darker and more aesthetically pleasing, she wants the total sweetness sum of the $N \times M$ platter to be negative. Finally, she does not want the judge to eat two canelés that taste the same in fear that it would bore him. This means that no two canelés in a single $A \times B$ subrectangle can have the same sweetness level. Can you assist Erin by telling her how to make the platter if possible?

Input Format:

The first and only line contains four space separated integers: N, M, A, and B. ($2 \leq A \leq N \leq 200$ and $2 \leq B \leq M \leq 200$)

Output Format:

Output "Yes" if it is possible for Erin to satisfy the requirements or "No" if otherwise.

If it is possible, output N lines of M integers each, representing the sweetnesses of the canelés she should make. Each integer should be between -10^9 and 10^9. Note that there may be more than one valid answer.

Sample Input:

```
5 4 3 2
```

Sample Output:

```
Yes
0 1 -5 -2
-8 -2 -1 0
-3 18 -4 15
3 0 -10 2
-6 -2 8 -9
```

§10 Cake Cutting

Chef Edward has baked an elaborate cake for his friend! The cake is placed on the coordinate plane, so locations in this problem are referenced by (x, y) coordinates. The cake is in the shape of a convex polygon with N vertices, labeled $V_1, V_2 \ldots, V_N$ in counterclockwise order. Edward has put frosting along the sides of the cake (the edges of the polygon), and nowhere else. He is going to put a single candle in the cake, then make a series of straight line cuts that divide the cake into polygonal pieces. The guests are picky, so each piece must satisfy the following conditions:

- Each vertex of each piece is a vertex of the original cake.
- Each piece that doesn't contain the candle has at most one side that does not have frosting (the piece that ends up with the candle may have more than one unfrosted side).

There are M places in the cake where Edward is considering putting the candle, given by $C_1, C_2, \ldots, C_M$. For each of these locations, Edward wants to know how many valid ways there will be to cut up the cake. Making no cuts and presenting the cake as one big piece counts as a way to cut the cake. Also, note that Edward may not cut along a line that contains the candle itself (the candle must lie strictly within a piece), and no cut should intersect a previous cut (except maybe at a vertex of the original cake). The order in which Edward performs the cuts doesn't matter.

Input Format:

The first line will contain two integers, N and M. $(1 \leq N, M \leq 10^3)$

The next N lines will each contain two integers, x_i and y_i. This coordinate pair represents V_i. The vertices will be given in counterclockwise order. $(0 \leq x_i, y_i \leq 10^9)$

The next M lines will each contain two integers, x_i and y_i. This coordinate pair represents C_i. $(0 \leq x_i, y_i \leq 10^9)$

Output Format:

Output in M separate lines the number of ways to cut the cake for each of the M possible candle locations. Since the answers may be very large, print them modulo $10^9 + 7$.

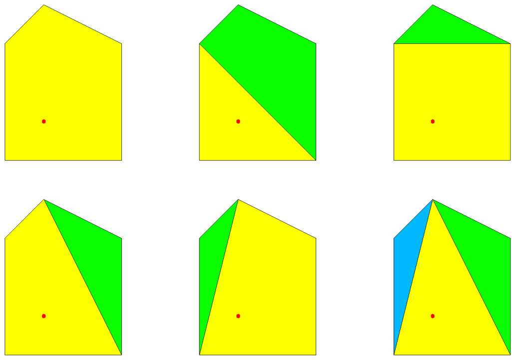
Sample Input:

```
5 2
0 0
3 0
3 3
1 4
0 3
1 1
1 3
```

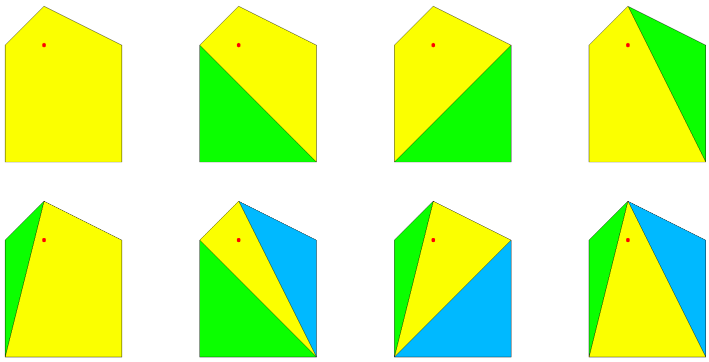
Sample Output:

```
6
8
```

There are 6 ways to cut the cake when the candle is at (1,1):



There are 8 ways to cut the cake when the candle is at (1,3):



§11 Lost Child

Little Gabe is wandering, lost in Billy Bonka's Chocolate Factory. Gabe's mother, Anna, is also wandering around in the factory, looking for him. Billy Bonka's advanced security system, which can describe the motion of anyone within the factory by their speed and the coordinate points of where they change direction, records Gabe's and Anna's movement.

Gabe's path can be described by N points $P_1, P_2, \ldots, P_N$ in the coordinate plane. Gabe travels at a constant speed of G units per second in straight line segments from P_1 to P_2, then P_2 to P_3, and so on until he reaches P_N . Similarly, Anna's path can be described by M points $Q_1, Q_2, \ldots, Q_M$, where she is moving at a speed of A units per second. Once each person reaches their final point, they stay there.

Billy Bonka was alerted that Gabe and Anna were still in the factory after closing time. He realizes that they got lost and wants to know the closest distance they ever were from each other. Since he is tired and wants to float around in his River of Chocolate, he has given you the information recorded by his security system to analyze. Can you find the distance for Billy Bonka?

Input Format:

The first line contains four integers: N, M, G, and A. ($1 \leq N, M \leq 10^5$ and $1 \leq G, A \leq 10^6$)

The next N lines will each contain two integers, x_i and y_i. This coordinate pair represents P_i. ($-10^6 \leq x_i, y_i \leq 10^6$)

The next M lines will each contain two integers, x_i and y_i. This coordinate pair represents Q_i. ($-10^6 \leq x_i, y_i \leq 10^6$)

Output Format:

Output a single number representing the closest Gabe and Anna ever get to each other. Round your answer to the nearest integer.

Sample Input:

```
3 4 6 7
0 0
30 0
0 20
30 10
20 -20
10 0
10 20
```

Sample Output:

```
3
```

The closest that Gabe ever gets to Anna is 3.443 units. This occurs after 9.415 seconds, when Gabe is at (7.960, 14.693) and Anna is at (10, 11.920). When rounded to the nearest integer, 3.443 is 3 (which is the final output).

§12 Outbreak

Professor Quarrel is studying bacteria in an attempt to develop a dessert that does not cause cavities. Currently, he has N species of bacteria. Each species is contained in a single petri dish in a line of N dishes labeled $1, 2, \ldots, N$. The amount of bacteria in dish i is given by A_i. Bacteria reproduce at different speeds, depending on their species. The growth constant of bacteria in dish i is B_i. Initially, amount of bacteria in each dish is equal to its growth constant ($A_i = B_i$ at the start).

Unfortunately, Quarrel's academic rival, Professor Snipe, is out to get him! At certain times, Snipe will feed bacteria in dishes $L, L+1, \ldots, R$ with sugar of sweetness X. Snipe puts more and more sugar in the dishes as he progresses down the line. As a result, after being fed, the bacteria in dish i multiply by their growth constant raised to a power proportional to the distance Snipe has already travelled. In other words, A_i becomes $A_i \cdot (B_i)^{X(i-L+1)}$ for all i in $L, L+1, \ldots, R$.

If the bacteria grow too strong, they will break out and ruin Quarrel's meticulous experiments. Quarrel is aware of Snipe's efforts and needs your help. To counteract Snipe's mischievous deeds, Quarrel will need some information about the bacteria. The strength of bacteria in dishes $L, L+1, \ldots, R$ is defined to be product of the numbers of bacteria in that interval, or $A_L \cdot A_{L+1} \cdot \ldots \cdot A_R$. Quarrel will need you to tell him the strength of certain intervals of bacteria throughout his experiments. You only need to find the remainder of the strength modulo $10^9 + 7$.

There will be two types of queries. Each query has a type T and left and right indices L and R. If $T = 1$, then it is a sabotage by Snipe, which will have an additional parameter X. Otherwise, if $T = 2$, it is a request by Quarrel. Note that L and R are always inclusive bounds on an interval of dishes, and L, R, and X can change between queries.

Input Format:

The first line contains two integers N and Q. ($1 \leq N, Q \leq 15,000$)

The second line contains N integers $B_1, \ldots, B_N$. ($1 \leq B_i \leq 10^9$)

The next Q lines contain three or four integers T, L, and R. ($1 \leq T \leq 2$ and $1 \leq L \leq R \leq N$)
If $T = 1$, then it is an update on dishes $L, L+1, \ldots, R$, and it has an additional parameter X. ($1 \leq X \leq 10^5$)
Otherwise $T = 2$, and Quarrel wants to know the strength of bacteria in the interval $L, L+1, \ldots R$.

Output Format:

For each query of type 2, output a single integer: the product of $A_L \cdot A_{L+1} \cdot \cdots \cdot A_R$ modulo $10^9 + 7$.

Sample Input:

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

Sample Output:

```
4860
233280000
```

Initially, A_i is equal to B_i for all i, and A looks like this:
1 2 3 4 5
After the first update, the array A looks like this:
1 8 243 4 5
The answer to the first request is $243 \cdot 4 \cdot 5 = 4860$.
After the second update, the array A looks this this:
1 8 729 64 625
The answer to the second request is $1 \cdot 8 \cdot 729 \cdot 64 \cdot 625 = 233280000$.