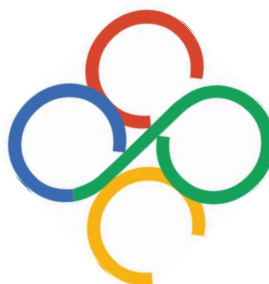


# China Collegiate Programming Contest Guangzhou Site

*CCPC 2021*

November 14



## Problems

- A. Math Ball
- B. Sweeping Robots
- C. Necklace
- D. Unnamed Easy Problem
- E. Matlab
- F. Cactus
- G. Slope
- H. Three Integers
- I. Pudding Store
- J. Cafeteria
- K. Magus Night
- L. Dynamic Convex Hull

*Do not open before the contest starts.*

## Problem A. Math Ball

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1 second  
Memory limit:         512 megabytes

Cake has  $n$  types of balls numbered from 1 to  $n$ , the  $i$ -th type of which has a sufficient number of indistinguishable balls and a cost  $c_i$ . Cake wants to take away some of them, but his backpack can only carry at most  $W$  balls. Simple knapsack problem cannot please Cake, so he wants to add some magic to the balls. Formally, if he takes  $k_i$  balls from the  $i$ -th type, the cost is  $C = k_1^{c_1} k_2^{c_2} \dots k_n^{c_n}$ . We want to know for all possible plans taking at most  $W$  balls, what the total cost modulo 998244353 is.

### Input

First line contains two integers  $n$  and  $W$ , and the second line contains  $n$  integers  $c_1, c_2, \dots, c_n$ , separated by space.

It is guaranteed that  $n \leq 10^5$ ,  $W \leq 10^{18}$ , and  $\sum c_i \leq 10^5$ .

### Output

Print the total cost modulo 998244353.

### Example

standard input	standard output
4 5 1 2 3 4	31

### Note

For the sample input, plans  $(k_1, k_2, k_3, k_4)$  with non-zero costs are  $(1, 1, 1, 2)$ ,  $(1, 1, 2, 1)$ ,  $(1, 2, 1, 1)$ ,  $(2, 1, 1, 1)$ , and  $(1, 1, 1, 1)$ , and the total cost is  $2^4 + 2^3 + 2^2 + 2^1 + 1 = 31$ .

## Problem B. Sweeping Robots

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            **3 seconds**  
Memory limit:         **1024 megabytes**

There are  $n$  sweeping robots in a line, and the  $i$ -th robot from left to right has a cost  $c_i$ . It is guaranteed that all  $c_i$ 's are distinct. The distance between the  $i$ -th robot and the  $(i+1)$ -th robot is  $d_i$ .

When you want to sweep the interval between the  $l$ -th robot and the  $r$ -th robot, you will first greedily choose the robot with the minimal cost in that interval. Then, the robot will find a path that covers every robot in that interval at least once, with a length as minimal as possible. Note that the robot does not need to return to the original place. Define the cost of interval  $[l, r]$  as the cost of the selected robot multiplies the length of the selected path.

You need to answer  $q$  queries, and each query is as follow: for a given interval  $[l, r]$ , what is the maximal cost over intervals  $[l', r']$  such that  $l \leq l' \leq r' \leq r$ ?

### Input

The first line consists of an integer  $n$  ( $1 \leq n \leq 5 \times 10^5$ ).

The second line consists of  $n$  integers, the  $i$ -th of which is  $c_i$  ( $1 \leq c_i \leq n$ ).

The third line consists of  $n - 1$  integers, the  $i$ -th of which is  $d_i$  ( $1 \leq d_i \leq 10^5$ ).

The fourth line consists of an integer  $q$  ( $1 \leq q \leq 10^6$ ).

The next  $q$  lines describes the queries, and the  $i$ -th line contains two integers  $l_i$  and  $r_i$  ( $1 \leq l_i \leq r_i \leq n$ ).

### Output

Print  $q$  integers, the  $i$ -th of which should be the answer for interval  $[l_i, r_i]$ .

### Example

standard input	standard output
5	16
4 1 5 2 3	2
2 8 4 2	12
4	12
3 5	
1 2	
2 4	
1 3	

## Problem C. Necklace

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            0.5 seconds  
Memory limit:         512 megabytes

Bob has given Alice a necklace as her birthday gift. That necklace has  $N$  crystals,  $M$  of which catches her fancy. Formally, crystals with labels of  $1..N$  are strung sequentially in a circle. And those catching her fancy are labelled  $n_1, n_2, \dots, n_M$ .

Alice would like the necklace to be divided into  $M$  pieces. She asked Bob to sever the necklace into  $M$  pieces. Each piece should consist of a sequence of crystals as in the original necklace (i.e., the crystal labels are contiguous, except that 1 following  $N$  is accepted) and should include one crystal catching Alice's fancy. Obviously, each crystal must belong to exactly one piece.

However, an excessively long piece of severed necklaces could not fit Alice's slim neck. So she wants to know, among all possible severings as requested, what the minimum length of the longest piece from a severed necklace is. She wants you to answer this question.

### Input

The first line of the input data contains 2 integers  $N$  and  $M$  ( $N < 10^{18}$  and  $M \leq 10^6$ ).

The second line contains  $M$  integers, the  $i$ -th of which represents  $n_i$ . The  $n_i$ 's are given in ascending order.

### Output

You need to output your answer in one line.

### Example

standard input	standard output
10 4 2 5 6 8	3

### Note

As for the example: You can sever the necklace into  $[1, 3]$ ,  $[4, 5]$ ,  $[6, 7]$ ,  $[8, 10]$ .

Considering huge scale of data, here is a way provided to help input faster:

```
#define gc() (is==it?it=(is=in)+fread(in,1,Q,stdin),(is==it?EOF:*is++):*is++)
const int Q=(1<<24)+1;
char in[Q],*is=in,*it=in,c;
void read(long long &n){
    for(n=0;(c=gc())<'0' || c>'9';);
    for(;c<='9'&& c>='0';c=gc())n=n*10+c-48;
}
```

## Problem D. Unnamed Easy Problem

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1 second  
Memory limit:         512 megabytes

Alice wants to compose a set of  $m$  easy problems. And she has  $n$  ideas: The first  $k$  ideas are extremely easy, while the other  $n-k$  are easy. In Alice's opinion, an easy problem is a set of ideas consisting of at least one easy idea and several extremely easy ideas. As we all know, two problems with the same set of ideas are not allowed, and the order of problems in a problem set does not matter. For each idea, Alice requires it to be exactly in an odd or even number of problems. Alice wants to know the total number of different possible problem sets, and you just need to output this number modulo 19260817.

### Input

The first line contains three positive integers  $n$ ,  $k$ , and  $m$  ( $1 \leq k \leq n \leq 10^9$ ,  $1 \leq m \leq 10^6$ , and  $2^n - 2^k \geq m$ ).

The second line contains two integers  $o_{ee}$  and  $o_{ez}$  ( $0 \leq o_{ee} \leq k$  and  $0 \leq o_{ez} \leq n - k$ ). It means  $o_{ee}$  extremely easy ideas and  $o_{ez}$  easy ideas are required to be in an odd number of problems, and the rest  $k - o_{ee}$  extremely easy ideas and  $n - k - o_{ez}$  easy ideas are required to be in an even number of problems.

### Output

The only line contains an integer — the total number mod 19260817.

### Example

standard input	standard output
4 2 2 0 1	4

### Note

In this example, ideas marked with 0, 1 are extremely easy, and the ones marked with 2, 3 are easy. Assume that the idea 2 is required to be in an odd number of problems, and the others are required to be in an even number of problems.

There are 12 possible easy problems. Each of them can be represented by a set of ideas:

$$\{2\}, \{3\}, \{2, 3\}, \{0, 2\}, \{0, 3\}, \{0, 2, 3\}, \{1, 2\}, \{1, 3\}, \{1, 2, 3\}, \{0, 1, 2\}, \{0, 1, 3\}, \{0, 1, 2, 3\}.$$

And there are 4 possible problem sets. Each of them can be represented by a set of sets above:

$$\{\{3\}, \{2, 3\}\}, \{\{0, 3\}, \{0, 2, 3\}\}, \{\{1, 3\}, \{1, 2, 3\}\}, \{\{0, 1, 3\}, \{0, 1, 2, 3\}\}.$$

## Problem E. Matlab

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            3 seconds  
Memory limit:         512 megabytes

The function  $f(x)$  is defined as the sum of all digits of  $x$  in hexadecimal. Given an  $n$ -digit hexadecimal number  $x$  and an index  $k$ , calculate

$$\sum_{i=0}^{x-1} f((16^k - 1) \cdot i) \bmod 2^{64}.$$

### Input

The first line contains two positive integers  $n$  and  $k$  ( $1 \leq k \leq n \leq 100$  and  $5k \geq n$ ).

The second line contains a string of length  $n$  — the value of given  $x$  in hexadecimal.

The string only consists of decimal digits and 'A', 'B', 'C', 'D', 'E', 'F'. Also the first digit is not '0'.

### Output

The only line contains an integer — the answer.

### Example

standard input	standard output
4 1 7FFF	1081320

## Problem F. Cactus

Input file:            standard input  
Output file:          standard output  
Time limit:           1 second  
Memory limit:        512 megabytes

An undirected connected graph is called a cactus, if and only if each edge of it belongs to at most one simple cycle. Now suppose there are  $f_n$  cactuses with  $n$  vertices, you need to calculate  $\sum_{i=1}^n \prod_{j \neq i} \frac{1+f_i-f_i f_j}{f_i-f_j}$ .

### Input

A single line contains an integer  $n$  ( $1 \leq n \leq 3 \times 10^5$ ).

### Output

Suppose the reduced form of the answer is  $\frac{x}{y}$ , and you only need to output the value of  $x \times y^{998244351} \bmod 998244353$ .

### Example

standard input	standard output
2	1

### Note

In the first example,  $f_1 = 1$ ,  $f_2 = 2$ , and the answer is 1.

## Problem G. Slope

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            8 seconds  
Memory limit:         512 megabytes

There are  $n$  points  $(x_i, y_i)$  in the plane, and all  $x_i$ 's are different.

You have to process  $q$  queries. Each query determines a subset of the  $n$  given points within the rectangle  $[x_l, x_r] \times [y_l, y_r]$ ; and the subset consists of all point  $i$  that satisfies  $x_l \leq x_i \leq x_r$  and  $y_l \leq y_i \leq y_r$ . The answer to the query is the minimal value of  $|\frac{y_i - y_j}{x_i - x_j}|$ , where  $i$  and  $j$  are different points in the subset described above and  $|a|$  is the absolute value of  $a$ . If the subset contains less than two points, please refer to the output section.

### Input

The first line contains two positive integers  $n$  and  $q$  ( $1 \leq n \leq 7 \times 10^3$  and  $1 \leq q \leq 7 \times 10^5$ ).

The  $i$ -th line of the next  $n$  lines contains two integers  $x_i$  and  $y_i$  ( $-10^9 \leq x_i, y_i \leq 10^9$ ).

In the next  $q$  lines, each contains four integers  $x_l, x_r, y_l$ , and  $y_r$  ( $-10^9 \leq x_l \leq x_r \leq 10^9$  and  $-10^9 \leq y_l \leq y_r \leq 10^9$ ) that constitute a query.

### Output

The output consists of  $q$  lines. The  $i$ -th line contains two non-negative coprime integers  $a$  and  $b$ , which represent the answer  $\frac{a}{b}$  to the  $i$ -th query. If  $a = 0$ , you should print '0 1'. If the subset contains less than two points, you should print '1 0'.

### Example

standard input	standard output
5 5	2 1
16 12	2 5
11 14	0 1
15 6	6 1
1 14	2 1
2 16	
1 2 13 17	
10 17 12 14	
1 17 11 19	
15 16 6 12	
11 15 1 19	



## Problem H. Three Integers

Input file:            **standard input**  
Output file:          **standard output**  
Time limit:           **1 second**  
Memory limit:        **512 megabytes**

You are given three non-negative integers  $a$ ,  $b$ , and  $c$ . Find three positive integers  $x$ ,  $y$ , and  $z$  that satisfy  $x \bmod y = a$ ,  $y \bmod z = b$ , and  $z \bmod x = c$ .

### Input

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

Each test case contains 3 integers  $a, b, c$  ( $0 \leq a, b, c \leq 10^9$ ) on a single line.

### Output

For each test case, if there are such three integers satisfying the condition, output “YES”, then output the three integers  $x, y, z$  ( $1 \leq x, y, z \leq 10^{18}$ ) on the following line, or “NO” otherwise.

### Example

standard input	standard output
4	YES
0 0 0	1 1 1
1 2 3	YES
6 6 6	5 2 8
11 3 3	NO
	YES
	11 45 14

## Problem I. Pudding Store

Input file:            standard input  
Output file:           standard output  
Time limit:            1 second  
Memory limit:         512 megabytes

159 is a boy. He has a pudding store.

There are  $n$  different puddings numbered from 1 to  $n$  lined up in a row in the pudding store. (Note that the  $i$ -th pudding in the row may not be the pudding with the number  $i$ .) Now,  $n$  students numbered from 1 to  $n$  are coming to sample these puddings in a specific way. That is, for the  $i$ -th student, he will sample each one of the first  $i$  puddings in the row. Sampling the pudding numbered  $i$  gives the sampler a satisfaction value of  $2 \times i$ . And if the sum of all satisfaction values that the  $i$ -th student gets is divisible by  $i$ , we would say that the  $i$ -th student is satisfied.

Now, 159 wants to know, how many different arrangements of the puddings in a row that every student will be satisfied after sampling. Two arrangements are different, if there exists a pudding that its position is different. Note that the number of arrangements may be very large so he just needs the remainder after division by 998244353 of the result.

### Input

Each test contains multiple test cases. The first line contains the number of test cases  $t$  ( $1 \leq t \leq 10$ ). Description of the test cases follows.

The first and only line of each test case contains one integer  $n$  ( $1 \leq n \leq 10^9$ ) — the number of the puddings and the students.

### Output

For each test case, print a single line that contains one integer — the number of satisfying arrangements modulo 998244353.

### Example

standard input	standard output
3	1
1	2
2	6
3	

## Problem J. Cafeteria

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1.5 seconds  
Memory limit:         512 megabytes

Class is over, and Dave is the first to rush into the cafeteria! There is a long table full of  $n$  dishes in order, and the  $i$ -th dish is  $a_i$ . Dave wants to eat  $m$  dishes, so he writes the name of the  $m$  dishes on a slip of paper, and the  $i$ -th dish is  $b_i$ . Dave and his classmates start lining up from the beginning of the long table, and he will take the dish in front of him to his plate when he finds it exactly the same as the first remaining dish on his slip of paper and then crosses it out from the slip. The cafeteria is so crowded that Dave would sometimes not see the dish in front of him. At the same time, it is clearly impossible for Dave to turn back and pick up a dish that he had already gone by.

This happens every day during the  $t$  days when Dave is in school, but the dishes that Dave wants to eat are always the same. However, on the  $i$ -th day in the crowded cafeteria, Dave can only see dishes  $l_i$  to  $r_i$  on the long table, and he wants to know the number of ways to satisfy his needs for each day. Two ways are considered different if there is at least one difference in the location where the dishes are taken.

### Input

The first line contains three integers  $n$  ( $1 \leq n \leq 2 \times 10^5$ ),  $m$  ( $1 \leq m \leq 30$ ), and  $t$  ( $1 \leq t \leq 10^6$ ).

The second line contains a string  $a_i$  of length  $n$ , and the third line contains a string  $b_i$  of length  $m$ . It is guaranteed that there are only lowercase characters.

In the next  $t$  lines, each line contains two positive integers  $l_i$  and  $r_i$  ( $1 \leq l_i \leq r_i \leq n$ ).

### Output

The output is large, so you only need to print 1 integer — The XOR sum of the  $t$  integers.

The  $i$ -th integer is the number of ways for Dave to take dishes on the  $i$ -th day, taken modulo  $10^9 + 7$ .

### Example

standard input	standard output
8 2 3 abcabbcb ab 1 5 1 7 3 8	5

### Note

In the sample:

For the first day, Dave have 3 ways to take dishes.

For the second day, Dave have 5 ways to take dishes.

For the third day, Dave have 3 ways to take dishes.

So the XOR sum of them is 5.

## Problem K. Magus Night

Input file:            standard input  
Output file:           standard output  
Time limit:            1 second  
Memory limit:         512 megabytes

Marisa Kirisame has been collecting Supernatural Beads. Now she has got  $n$  Beads, and she wants to use them to perform her representative magic, Master Spark.

When she prepares to perform magic, the Beads will be energized. Formally, each of the Beads will get an integer in  $[1, m]$  as its *energy*, completely **randomly** and **independently**. Note that the *energy* of a Bead may change when she repeats energizing.

After some practising, she finds a few interesting facts about her Beads. Assume that the *energy* of the Beads is  $s_1, s_2, \dots, s_n$ , then the *power* of the magic is  $\text{lcm}(s_1, s_2, \dots, s_n)$ , and the *cost* of the magic is  $\text{gcd}(s_1, s_2, \dots, s_n)$ . Here  $\text{lcm}(s_1, s_2, \dots, s_n)$  denotes the *Least Common Multiple (LCM)* of integers  $s_1, s_2, \dots, s_n$ , and  $\text{gcd}(s_1, s_2, \dots, s_n)$  denotes the *Greatest Common Divisor (GCD)* of integers  $s_1, s_2, \dots, s_n$ .

Marisa is strict about magic. She sets two positive integers  $p$  and  $q$  no greater than  $m$ , and she regards a magic *successful* if and only if the *power* of the magic is **no less** than  $p$  and the *cost* of the magic is **no greater** than  $q$ . She thinks the *value* of a magic is  $\prod_{i=1}^n s_i$  if the magic is *successful*, and 0 otherwise.

Now she wonders about the expected *value* of magic. Let the answer be  $x$ , it is obvious that  $x \cdot m^n$  is an integer. As this number may be large, you only need to tell her  $x \cdot m^n \bmod 998244353$ .

### Input

The only line of the input contains four integers,  $n$ ,  $m$ ,  $p$ , and  $q$  — the number of the Beads, the maximum possible *energy* of the Beads and the two parameters on defining *successful* magics.

It is guaranteed that  $1 \leq n \leq 998244351$  and  $1 \leq p, q \leq m \leq 2 \times 10^5$ .

### Output

The only line of the output contains a single integer in  $[0, 998244353)$  — the expected *value* of a magic, that is,  $x \cdot m^n$  in the last paragraph of the statement.

### Examples

standard input	standard output
2 4 3 2	66
7 2 1 2	2187

### Note

In the first example, there are 16 possible situations, and the *successful* ones are listed below:

$\{1, 3\}$ ,  $\{1, 4\}$ ,  $\{2, 3\}$ ,  $\{2, 4\}$ ,  $\{3, 1\}$ ,  $\{3, 2\}$ ,  $\{3, 4\}$ ,  $\{4, 1\}$ ,  $\{4, 2\}$ ,  $\{4, 3\}$ .

Those situations are not *successful*, for their *power* is less than 3:

$\{1, 1\}$ ,  $\{1, 2\}$ ,  $\{2, 1\}$ ,  $\{2, 2\}$ .

Those situations are not *successful*, for their *cost* is more than 2:

$\{3, 3\}$ ,  $\{4, 4\}$ .

In the second example, it is obvious that all  $2^7$  situations are *successful*.

## Problem L. Dynamic Convex Hull

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            10 seconds  
Memory limit:         512 megabytes

There are  $N$  points.

You will be given  $M$  queries. Each query will give you several points. You must answer the area of the convex hull of these points merged with the initial  $N$  points.

### Input

This problem contains multiple test cases.

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

For each test case:

The first line contains an integer  $N$  ( $3 \leq N \leq 2 \times 10^5$ ) — the number of the initial points.

Then  $N$  lines follow. The  $i$ -th of them contains two integers  $x_i, y_i$  ( $|x_i|, |y_i| \leq 10^9$ ) — the coordinate of the  $i$ -th initial point.

The next line contains an integer  $M$  ( $1 \leq M \leq 2 \times 10^5$ ) — the number of queries.

Then  $M$  queries follow. For each query: the first line contains an integer  $k$  ( $1 \leq k \leq 10$ ) — the number of points in this query; then  $k$  lines follow, the  $i$ -th of which contains two integers  $x_i, y_i$  ( $|x_i|, |y_i| \leq 10^9$ ) — the coordinate of the  $i$ -th point in this query.

It is guaranteed that in all test cases, the sum of  $N$  is no more than  $10^6$ , and the sum of  $k$  is no more than  $10^6$ .

### Output

For each query in each test case, output  $2 \times S$  in one line.  $S$  indicates the area in this query.

It can be proved that  $2 \times S$  is always an integer.

### Example

standard input	standard output
1	39
8	
-1 2	
-2 1	
-2 -1	
-1 -2	
1 -2	
2 -1	
2 1	
1 2	
1	
3	
0 3	
0 4	
1 5	