Problem A. Cookies

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

You are to hold a cookie party! You have prepared N cookies numbered from 1 through N. The **sweetness** of the cookie i is A_i. You expect that M children numbered from 1 to M will attend the party. All of them will bring their homemade cookies, and the child i will bring a cookie of sweetness B_i. Besides, you know the taste preference of each child. The child i loves sweet cookies if $S_i =$`S` and loves bitter cookies if $S_i =$`B`.

The party will proceed in the following manner:

- First, you are given an integer k and put cookies $1, 2, \cdots, k$ on the table.
- Then, children $1, 2, \cdots, M$, in this order, come to the table. When the child i comes to the table, the child first put his/her homemade cookie on the table. Then, if the child loves sweet cookies, he/she eats the sweetest cookie (a cookie with the largest sweetness) on the table. If the child loves bitter cookies, he/she eats the bitterest cookie (a cookie with the smallest sweetness) on the table. Note that each child eats exactly one cookie, and a child may eat his/her cookie.
- Finally, you eat all the cookies left on the table.

You have not yet decided the value of k. For each integer $k = 1, 2, \cdots, N$, find the sum of the sweetness of cookies you will eat.

Note that only after you get the answer for $k = i$ can you know the value of A_{i+1}. See the input section for more details.

Input

Input is given from Standard Input in the following format:

N

$A'_1\ A'_2\ \cdots\ A'_N$

M

$B_1\ B_2\ \cdots\ B_M$

S

Here A'_i is the encrypted value of A_i, and the real value can be calculated as $A_i = (A'_i + lastans \mod 10^9)$, where $lastans$ denotes the answer for $k = i - 1$ if $i > 1$ and 0 if $i = 1$.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq A_i \leq 10^9 - 1$
- $0 \leq A'_i \leq 10^9 - 1$ (see the input section for the definition of this variable)
- $1 \leq M \leq 2 \times 10^5$
- $0 \leq B_i \leq 10^9 - 1$
- $|S| = M$
- S_i is either '`S`' or '`B`'.

- All values in input are integers.

Output

Print N integers in one line. The i-th integer should be the sum of the sweetness of cookies you will eat when $k = i$.

Examples

standard input

```
3
3 999999999 0
2
4 2
BS
```

standard output

```
2 5 9
```

standard input

```
10
810737462 262894941 12979345 90139935 834123271 768745833 928886601 144082546 35547099 840309069
10
854737038 93768450 848842263 62613614 800833082 316988396 203584286 283415773 762732633 756024517
SBSSSBSSBS
```

standard output

```
756024517 959608803 1243024576 1560012972 1893177483 2287313726 2503514053 3151110652 3337768403 3515845875
```

Note

In first sample, $A = (3, 1, 5)$.

When $k = 2$, the party proceeds as follows:

- You put 2 cookies of sweetness 3 and 1.
- The child 1 puts the cookie of sweetness 4 and eats the cookie of sweetness 1.
- The child 2 puts the cookie of sweetness 2 and eats the cookie of sweetness 4.
- You eat cookies of sweetness 2 and 3.

Problem B. Evacuation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	1024 mebibytes

There are $N+2$ towns in a straight line. The towns are numbered from 0 through $N+1$ from left to right. Each town i ($1 \le i \le N$) has a shelter which can contain up to A_i people.

Currently, S people are traveling together and visiting one of the towns. However, you don't know which town they are visiting.

You just got to know that there are Q meteorites that can hit the towns. The i-th meteorite may strike towns $L_i, L_i+1, \cdots, R_i$. To ensure the safety of the travelers, for each meteorite, you want to calculate the **evacuation cost**.

The evacuation cost for a meteorite is calculated as follows:

- The evacuation cost is the minimum total cost needed to make all travelers **safe** no matter which town they are visiting.
- A person is safe if he/she is in a shelter or a town outside the effect of the meteorite.
- It takes 1 unit cost to move one person to an adjacent town (two towns are adjacent iff their numbers differ by exactly 1).

Note that only moving people costs money, and other actions (like accommodating people in a shelter) don't. It is guaranteed that towns 0 and $N+1$ will never be affected by meteorites, so it is always possible to make all travelers safe.

Write a program that, for each meteorite, calculates the evacuation cost for that meteorite.

Input

Input is given from Standard Input in the following format:

N S

A_1 A_2 $\cdots$ A_N

Q

L_1 R_1

L_2 R_2

$\vdots$

L_Q R_Q

Constraints:

- $1 \le N \le 2 \times 10^5$
- $1 \le S \le 10^{12}$
- $0 \le A_i \le S$
- $A_1 + A_2 + \cdots + A_N \le 10^{12}$
- $1 \le Q \le 2 \times 10^5$
- $1 \le L_i \le R_i \le N$

- All values in input are integers.

Output

Print Q lines. The i-th line should contain the evacuation cost for the meteorite i.

Example

standard input	standard output

5 10
3 1 1 4 1
3
1 4
3 5
2 5
	14
	10
	13

Note

- For the first meteorite, it takes 14 unit costs when the travelers are visiting the town 2.
- For the second meteorite, it takes 10 unit costs when the travelers are visiting the town 4.
- For the third meteorite, it takes 13 unit costs when the travelers are visiting the town 3.

Problem C. Sum Modulo

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	1024 mebibytes

Snuke found a random number generator. It generates an integer between 1 and N (inclusive). An integer sequence $A_1, A_2, \cdots, A_N$ represents the probability that each of these integers is generated. The integer i ($1 \le i \le N$) is generated with probability A_i/S, where $S = \sum_{i=1}^{N} A_i$. The process of generating an integer is done independently each time the generator is executed.

Snuke has an integer X, which is now 0. He can perform the following operation any number of times:

- Generate an integer v with the generator and replace X with $X + v \mod M$.

Find the expected number of operations until X becomes K, and print it modulo 998244353. More formally, represent the expected number of operations as an irreducible fraction P/Q. Then, there exists a unique integer R such that $R \times Q \equiv P \mod 998244353,\ 0 \le R < 998244353$, so print this R.

We can prove that the expected number of operations until X becomes K is a finite rational number. However, we did **not** prove its integer representation modulo 998244353 can be defined. Our sincerest apologies. Nonetheless, you don't have to worry about division by 0. More precisely, We can model this problem as an absorbing markov chain (`https://en.wikipedia.org/wiki/Absorbing_Markov_chain`), and we guarantee that in all tests, the corresponding fundamental matrices can be defined modulo 998 244 353.

Input

Input is given from Standard Input in the following format:

$N\ M\ K$

$A_1\ A_2\ \cdots\ A_N$

Constraints

- $1 \le N \le \min(500, M-1)$
- $2 \le M \le 10^{18}$
- $1 \le K \le M-1$
- $1 \le A_i \le 100$
- All values in input are integers.

Output

Output the expected number of operations until X becomes K, modulo 998 244 353.

Example

standard input	standard output
2 3 1 1 1	2
10 100 50 1 2 3 4 5 6 7 8 9 10	439915532

Problem D. Xor Sum

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Determine whether there exists a sequence of N nonnegative integers $a_1, a_2, \cdots, a_N$ that satisfies all of the following conditions, and if it exists, find the minimum possible value of the maximum of the array.

- $a_1 + a_2 + \cdots + a_N = S$
- $a_1 \oplus a_2 \oplus \cdots \oplus a_N = X$ (Here $\oplus$ denotes bitwise xor operation)

Note that there are T tests in one input file.

Input

Input is given from Standard Input in the following format:

T

N_1 S_1 X_1

N_2 S_2 X_2

$\vdots$

N_T S_T X_T

Here, N_i, S_i, X_i represent values of N, S, X for the i-th test, respectively.

Constraints:

- $1 \leq T \leq 500$
- $1 \leq N \leq 2^{60} - 1$
- $0 \leq S \leq 2^{60} - 1$
- $0 \leq X \leq 2^{60} - 1$
- All values in input are integers.

Output

Print T lines. In the i-th line, print -1 if there doesn't exist an array with the mentioned property in the i-th test, and print the minimum possible value of the maximum of the array if it exists.

Example

standard input	standard output

6
3 9 3
4 8 0
6 19 1
1 15 15
2 6 5
5 4 3
	3
	2
	4
	15
	-1
	-1

Note

The following is a solution for each test:

- (3,3,3)
- (2,2,2,2)
- (2,3,3,3,4,4)
- (15)
- Impossible
- Impossible

Problem E. Count Modulo 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3.5 seconds
Memory limit:	1024 mebibytes

You are given K distinct nonnegative integers $A_1, A_2, \cdots, A_K$. Count the number of sequences of N nonnegative integers $a_1, a_2, \cdots, a_N$ that satisfies all of the following conditions, modulo **2**.

- $a_1 + a_2 + \cdots + a_N = S$
- For each i ($1 \leq i \leq N$), there exists an integer j such that $a_i = A_j$.

Note that there are T tests in one input file.

Input

Input is given from Standard Input in the following format:

T

Description of the 1-st test

Description of the 2-nd test

$\vdots$

Description of the T-th test

The description of each test is in the following format:

N S K

A_1 A_2 $\cdots$ A_K

Constraints:

- $1 \leq T \leq 5$
- $1 \leq N \leq 10^{18}$
- $0 \leq S \leq 10^{18}$
- $1 \leq K \leq 200$
- $0 \leq A_1 < A_2 < \cdots < A_K \leq 10^5$
- All values in input are integers.

Output

For each test, print the count modulo 2.

Example

standard input

2
5 10 3
1 2 3
1000000000000000000 254533217712393810 10
0 1683 21728 31623 35054 37834 39329 56842 68603 74742

standard output
1 0

Note

In the first test, there are a total of 51 sequences that satisfy conditions.

Problem F. Robots

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 mebibytes

There are N robots numbered from 1 through N and N antennas numbered from 1 through N in a straight line. The coordinate of the robot i is a_i and the coordinate of the antenna i is b_i. All coordinates are distinct.

Currently, all antennas are inactive. You are going to activate them one by one. When you activate an antenna, the nearest robot (if two robots are closest to it, only the left one) moves to the antenna and explodes along with it.

Find an order to activate antennas so that the total distance of robots' moves is minimum possible.

Input

Input is given from Standard Input in the following format:

N

$a_1\ a_2\ \ldots\ a_N$

$b_1\ b_2\ \ldots\ b_N$

Constraints:

- $1 \le N \le 2 \times 10^5$
- $0 \le a_1 < a_2 < \cdots < a_N \le 10^9$
- $0 \le b_1 < b_2 < \cdots < b_N \le 10^9$
- $a_1, a_2, \ldots, a_N, b_1, b_2, \ldots, b_N$ are all distinct.
- All values in input are integers.

Output

Print the answer in the following format:

X

$p_1\ p_2\ \ldots\ p_N$

Here, X must be a minimum total distance, and p_i is the index of the antenna that you activate in the i-th.

If there are multiple solutions, you can print any of them.

Example

standard input	standard output

3
1 2 3
11 12 13
	30
	3 2 1

Problem G. Matrix Inversion

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 mebibytes

You have an N by N grid board, which is initially empty. You will write an integer to each cell, using each integer from 1 to N^2 exactly once. Let $M_{i,j}$ be the integer written on the cell in the i-th row from the top and the j-th column from the left.

Let's define sequences A and B as follows:

- $A = M_{1,1}, M_{1,2}, \ldots, M_{1,N}, M_{2,1}, M_{2,2}, \ldots, M_{2,N}, \ldots, M_{N,N}$
- $B = M_{1,1}, M_{2,1}, \ldots, M_{N,1}, M_{1,2}, M_{2,2}, \ldots, M_{N,2}, \ldots, M_{N,N}$

For example, when the board looks like this,

```
1 3 4
2 7 6
9 8 5
```

A and B are defined as follows.

- $A = 1, 3, 4, 2, 7, 6, 9, 8, 5$
- $B = 1, 2, 9, 3, 7, 8, 4, 6, 5$

You are given integers N, X, Y. Find a way to fill in the cells so that the inversion numbers of A and B are X and Y respectively, or report that it is impossible to do so.

Note: an inversion number of a sequence $C = c_1, c_2, \ldots, c_{N \times N}$ is the number of pairs (i, j) s.t. both $i < j$ and $c_i > c_j$ are satisfied.

Input

Input is given from Standard Input in the following format:

$N\ X\ Y$

Constraints:

- $2 \leq N \leq 300$
- $0 \leq X, Y \leq \frac{N^2(N^2-1)}{2}$

Output

If there is no solution, print '`No`'.

Otherwise, print the answer in the following format:

Yes

$M_{1,1}\ M_{1,2}\ \ldots\ M_{1,N}$

$M_{2,1}\ M_{2,2}\ \ldots\ M_{2,N}$

$\vdots$

$M_{N,1}\ M_{N,2}\ \ldots\ M_{N,N}$

If there are multiple solutions, you can print any of them.

Examples

standard input	standard output
3 8 13	Yes 1 3 4 2 7 6 9 8 5

Problem H. Construct Points

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 mebibytes

Construct four points a, b, c and d on the two-dimensional plane that satisfy the following conditions:

- x and y coordinates of all points are integers, and their absolute values don't exceed 10^9
- a and b are different, and c and d are different.
- Let l be the line passing through a and b, and let m be the line passing through c and d.

Then,

- l and m are not parallel.
- the absolute values of x and y coordinates of the cross point of l and m are not less than 10^{27}.

Input

There is no input.

Output

Print the answer in the following format:

a_x a_y

b_x b_y

c_x c_y

d_x d_y

Problem I. Amidakuji

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are given a positive integer N. Construct a sequence of permutations of $(1, 2, \cdots, N)$, $p_1, p_2, \ldots, p_K$, that satisfy following conditions, or report that it's impossible.

- $0 \leq K \leq \lceil \log_2 N \rceil + 1$, where K is the length of the sequence.
- $p_1, p_2, \ldots, p_K$ are permutations of $(1, 2, \ldots, N)$.

In other words, they are bijections from
$1, 2, \ldots, N$ to
$1, 2, \ldots, N$.

- For all x and y $(1 \leq x, y \leq N)$, there is a sequence of bijections $q_1, q_2, \ldots, q_K$ such that $(q_K \circ q_{K-1} \circ \cdots \circ q_1)(x) = y$ and $q_i = p_i$ or p_i^{-1} for all i.

Here, $\circ$ denotes function composition, and when $K = 0$, $q_K \circ q_{K-1} \circ \cdots \circ q_1$ is defined as an identity function.

Input

Input is given from Standard Input in the following format:

N

Constraints:

$1 \leq N \leq 1000$

Output

If there is no solution, print -1. Otherwise, print the answer in the following format:

K

$p_{1,1}\ p_{1,2}\ \cdots\ p_{1,N}$

$\vdots$

$p_{K,1}\ p_{K,2}\ \cdots\ p_{K,N}$

Here, $p_{i,j}$ must be a value of $p_i(j)$.

If there are multiple solutions, you can print any of them.

Example

standard input	standard output

3	3
	1 3 2
	2 3 1
	3 1 2

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

Note

In Sample 1 for $x = 2, y = 1$, we can set $q_1 = p_1, q_2 = p_2^{-1}, q_3 = p_3$ and get $q_3(q_2(q_1(2))) = 1$.

Problem J. Median Replace Hard

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

Do you know this problem: `https://atcoder.jp/contests/agc022/tasks/agc022_e`? We generalize it a bit.

You got a binary string $P = P_0P_1P_2P_3P_4P_5P_6P_7$ of length 8.

You think a binary string X of odd length N is beautiful if it is possible to apply the following operation $\frac{N-1}{2}$ times so that the only character of the resulting string is '`1`' :

- Choose three consecutive bits of X, (X_i, X_{i+1}, X_{i+2}), and replace them by the $(X_i + 2X_{i+1} + 4X_{i+2})$-th bit of P.

Note that, when $P = 00010111$, this definition is the same as the original AGC problem.

You have a string S consisting of characters '`0`', '`1`', and '`?`'. You want to know the number of ways to replace the question marks with '`1`' or '`0`' so that the resulting string is beautiful, modulo $10^9 + 7$.

Note that there are T tests in one input file.

DO NOT submit your solution to the original problem during the contest. If you want to check your answer, please use AtCoder's Testcases: `https://www.dropbox.com/sh/nx3tnilzqz7df8a/AAAYlTq2tiEHl5hsESw6-yfLa?dl=0`.

Input

Input is given from Standard Input in the following format:

T

P S

P S

$\vdots$

P S

Constraints:

- $1 \le T \le 256$
- $|P| = 8$
- $1 \le |S|$ and $\sum |S| \le 300,000$
- $|S|$ is odd.
- All characters of P are either '0' or '1'.
- All characters of S are either '0', '1', or '¿.

Output

For each case, print the answer in a line.

Example

standard input
7 00010111 1??00 00010111 ? 00010111 ?0101???10???00?1???????????????0????????????1????0 01010101 1???? 01010101 0???? 00001111 1???? 00001111 0????
standard output
2 1 402589311 16 0 8 8

Problem K. Game and Queries

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Alice and Bob are going to play a game. The rule of the game is as follows:

- There are some monsters, each of which has a specified HP.
- Alice and Bob take turns alternately, with Alice going first.
- In Alice's turn, she chooses one of the monsters and increases its HP by 1.
- In Bob's turn, he chooses one of the monsters and decreases its HP by 2. If the monster's HP becomes 0 or less, the monster disappears.
- The game ends when k monsters disappear.

Alice's objective is to finish the game as late as possible, while Bob's is as soon as possible.

Initially, there are no monsters. You have to process Q queries of the following types:

- Type 1: You are given two integers X_i and Y_i. After this query, the number of monsters whose HPs are X_i becomes Y_i.
- Type 2: Given an integer K_i, calculate how many turns would Bob take if the game started with current monsters and $k = K_i$, assuming both players play optimally.

Note that the game doesn't happen in reality, and the monsters don't disappear.

Input

Input is given from Standard Input in the following format:

Q

Description of the 1-st query

Description of the 2-nd query

$\vdots$

Description of the Q-th query

The description of each query is in one of the following formats:

Type 1: 1 X_i Y_i

Type 2: 2 K_i

Constraints:

- $1 \le Q \le 3 \times 10^5$
- $1 \le T_i \le 2$
- $1 \le X_i \le 10^6$
- $0 \le Y_i \le 10^6$
- $1 \le K_i \le$ (the number of current monsters)

- There is at least one query of the type 2
- All values in input are integers.

Output

For each query of the type 2, print the answer in a line.

Example

standard input	standard output

6
1 1 4
2 3
1 2 3
2 6
1 2 2
2 6
	3
	7
	8

20
1 1 12
2 12
1 2 15
2 12
2 3
1 12 10
2 27
1 14 6
2 7
2 43
2 22
1 8 7
1 1 11
2 49
1 5 19
2 38
2 8
1 12 14
1 16 1
2 24
	12
	12
	3
	42
	7
	246
	25
	301
	91
	8
	32

Note

After the 5-th query, there are 4 monsters whose HPs are 1 and 2 monsters whose HPs are 2.

Problem L. Yosupo's Algorithm

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 mebibytes

> ...Can you replicate my master thesis in 5 hours?
>
> Yosupo

You are given N red points and N blue points on a two dimensional plane. The i-th red point's coordinate is (r_i^x, r_i^y), and its weight is r_i^w. The i-th blue point's coordinate is (b_i^x, b_i^y), and its weight is b_i^w.

Process Q queries. In the i-th query, you are given two integers L_i and R_i, and choose a red point j and a blue point k with following conditions:

- $r_j^y < b_k^y$
- $(r_j^x < L_i$ and $R_i < b_k^x)$ or $(L_i < r_j^x$ and $b_k^x < R_i)$

Your task is to maximize the sum of weights of the two points or report that it is impossible to select two points.

Input

Input is given from Standard Input in the following format:

N
r_1^x r_1^y r_1^w
$\vdots$
r_N^x r_N^y r_N^w
b_1^x b_1^y b_1^w
$\vdots$
b_N^x b_N^y b_N^w
Q
L_1 R_2
$\vdots$
L_Q R_Q

Constraints:

- $1 \le N \le 100{,}000$
- $1 \le Q \le 500{,}000$
- $-1{,}000{,}000{,}000 \le r_i^x, L_i \le -1$
- $1 \le b_i^x, R_i \le 1{,}000{,}000{,}000$
- $1 \le r_i^y, b_i^y \le 1{,}000{,}000{,}000$

- $1 \leq r_i^w, b_i^w \leq 1{,}000{,}000{,}000$
- $r_1^x, \cdots, r_N^x, b_1^x, \cdots, b_N^x, L_1, \cdots, L_Q, R_1, \cdots, R_Q$ are all distinct
- $r_1^y, \cdots, r_N^y, b_1^y, \cdots, b_N^y$, are all distinct

Output

For each query, in a line, print the maximum sum of weights of the selected points, or -1 if it is impossible to choose two points.

Examples

standard input	standard output

2
-3 1 1
-6 3 10
3 4 100
5 2 1000
5
-5 4
-2 6
-4 1
-10 10
-1 2
	101
	-1
	110
	1001
	1001

10
-389 786 414303478
-159 301 976196121
-268 599 754785437
-605 652 597104844
-199 841 214521748
-192 8 581825989
-515 898 509582353
-297 36 854072992
-489 616 41481895
-665 876 378086770
869 583 376652629
509 222 380009514
354 693 428231281
519 738 608396032
100 811 220629740
960 708 928349711
324 89 710139852
716 897 771429659
647 203 72269757
368 699 540753047
10
-350 499
-956 639
-287 504
-915 742
-777 135
-176 487
-150 987
-133 10
-852 147
-476 106
	1564212844
	1584592153
	1782422703
	1747625780
	1196825861
	1782422703
	-1
	1904545832
	1196825861
	1525454555