Convolution (ARRQRY)

You are given two integer sequences $A_1, A_2, \ldots, A_N$ and $B_1, B_2, \ldots, B_M$. For any two sequences $U_1, U_2, \ldots, U_p$ and $V_1, V_2, \ldots, V_q$, we define

$$Score(U,V) = \sum_{i=1}^{p} \sum_{j=1}^{q} U_i \cdot V_j \, .$$

You should process Q queries of three types:

- $1 \; L \; R \; X$: Add X to each of the elements $A_L, A_{L+1}, \ldots, A_R$.
- $2 \; L \; R \; X$: Add X to each of the elements $B_L, B_{L+1}, \ldots, B_R$.
- 3: Print $Score(A, B)$ modulo $998,244,353$.

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first line of each test case contains two integers, N and M, denoting the length of A and B respectively.
- The second line contains N integers, elements of A.
- The third line contains M integers, elements of B.
- The next line will contain an integer, Q, number of queries.
- Each of the next Q lines will contain one of 3 kinds of updates as mentioned in the statement

It's guaranteed that each update is a valid update operation.

Output

For each query of the third type, print a single line containing one integer - the answer to that query.

Constraints

- $1 \leq T \leq 10$
- $2 \leq N, M, Q \leq 10^5$
- $0 \leq |A_i|, |B_i|, |X| \leq 10^5$

Example Input

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

Example Output

```
72
24
90
```

Explanation

Before the first operation, $A = [2, -1, 5],\ B = [3, 3, 2, 4]$

So, for the first operation, $Score(A,\ B) = 2*3+2*3+2*2+2*4\ +(-1)*3\ +(-1)*3$ $+(-1)*2 + (-1)*4\ +5*3\ +5*3\ +5*2\ +5*4 = 72.$

After the second query $A = [2, -3, 3]$, $B = [3, 3, 2, 4]$

So, for the third query, $Score(A, B) = 2*3+2*3+2*2\ +2*4\ +(-3)*3\ +(-3)*3$ $+(-3)*2\ +(-3)*4\ +3*3\ +3*3\ +3*2\ +3*4 = 24.$

Beauty and The Array (BEAUAR)

Let’s define a function score:

$$Score(p[1...m], q[1...m]) = \sum_{i=1}^{m} (m - i + 1) * p[i] + q[i]$$

Let’s define Beauty of an array:

$$Beauty(p[1...n], q[1...n]) = max_{1 \le i \le j \le n} Score(p[i...j], q[i...j])$$

Here $p[i...j]$ means sub-array of p starting from i-th position and ending at j-th position.

You are given two arrays A and B of n integers. You can do the following operation any number of times, possibly zero.

- Perform a left circular rotation on **both** of the arrays simultaneously i.e move the first element of each array to the last position and shift all remaining elements one position left. For example, If an array is $\{1, 5, 3, 2\}$ and you perform left circular rotation once, then the array will become $\{5, 3, 2, 1\}$. If you perform the operation again, the array will become $\{3, 2, 1, 5\}$.

You need to maximize the value $Beauty(A, B)$.

Please note that, in one operation, you can't perform a left circular rotation on only one array, you must perform it on both of them.

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- Each test case consists of 3 lines:
  - First line contains n, number of elements of the array.
  - Next line contains n space separated integers, denoting elements of the array A.
  - Next line contains n space separated integers, denoting elements of the array B.

Output

For each testcase, output in a single line answer for that test case, which is the maximum $Beauty(A, B)$ that can be achieved.

Constraints

- $1 \le T \le 10$
- $1 \le n \le 2^{17}$
- $0 \le |A_i|, |B_i| \le 2^{17}$

Example Input

```
1
5
-1 3 -8 -1 4
3 1 -2 3 -10
```

Example Output

```
7
```

Explanation

In the sample test, to get the maximum beauty value (7), you have to perform the left circular operation 2 times. Then the arrays will be :

- $A = \{-8, -1, 4, -1, 3\}$
- $B = \{-2, 3, -10, 3, 1\}$.

If we now pick $i = 3, j = 5$, we get $Score(A[3...5], B[3...5]) = 7$.
This is the maximum that we can obtain. So, $Beauty(A, B) = 7$

IPL Begins (FCIPL)

This year p footballers and q cricketers have been invited to participate in IPL (Indian Programming League) as guests. You have to accommodate them in r rooms such that-

- No room may remain empty.
- A room may contain either only footballers or only cricketers, not both.
- No cricketers are allowed to stay alone in a room.

Find the number of ways to place the players. Note though, that all the rooms are identical. But each of the cricketers and footballers are unique.

Since the number of ways can be very large, print the answer modulo $998,244,353$.

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first and only line of each test case contains three space-separated integers p, q and r denoting the number of footballers, cricketers and rooms.

Output

For each test case, output the number of ways to place the players modulo $998,244,353$.

Constraints

- $1 \leq T \leq 100$
- $1 \leq p, q, r \leq 100$

Example Input

```
4
2 1 4
2 4 4
2 5 4
2 8 4
```

Example Output

```
0
3
10
609
```

Explanation

Example case 2: Three possible ways are:

- {Footballer 1}, {Footballer 2}, {Cricketer 1, Cricketer 2}, {Cricketer 3, Cricketer 4}
- {Footballer 1}, {Footballer 2}, {Cricketer 1, Cricketer 3}, {Cricketer 2, Cricketer 4}
- {Footballer 1}, {Footballer 2}, {Cricketer 1, Cricketer 4}, {Cricketer 2, Cricketer 3}

Please note that the rooms are identical.

Is This a Give Away (GVAWAY)

You are given two integers l and r.
You have to choose k **real** numbers in the interval $[l, r]$ uniform randomly.
What is the expected count of distinct numbers chosen by you?

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first and only line of each test case contains three space-separated integers l, r and k.

Output

For each test case, print a single line containing one integer - the expected count of distinct numbers chosen. It can be proved that the expected count is always an integer.

Constraints

- $1 \le T \le 10^5$
- $1 \le k \le 100$
- $1 \le l \le r \le 100$

Example Input

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

Example Output

```
4
1
2
```

Explanation

Example case 2: You choose only 1 real number, so it doesn't matter which real number you chose, number of distinct number is always 1. So expected count of distinct number is 1.

K-Prob (KPRB)

You are given a string S of length n and an integer K.
We will define $F(L)$ here. Let's choose K random sub-strings of length L from S.
Then $F(L)$ equals the probability such that all of them are pairwise identical.

See the **Explanation** section for more clarity.

You have to find $F(L)$ for each $1 \le L \le n$.

It can be shown that any $F(L)$ can be expressed as a fraction $\frac{P}{Q}$, where P and Q are coprime integers, $P \ge 0$, $Q > 0$ and Q is coprime with 998244353. You should compute $P \times Q^{-1}$, where Q^{-1} denotes the multiplicative inverse of Q modulo 998244353.

Input:

- First line will contain two integers, n (the length of string S) and K.
- Next line will contain the string S.

Output:

Print n space separated integers, $F(L)$, for each $1 \le L \le n$, in a single line,

Constraints

- $1 \le n \le 10^6$
- $2 \le K \le 10^6$
- It's guaranteed that S contains only lower-case Latin letters.

Sample Input:

```
5 3
ababa
```

Sample Output:

```
239578645 748683265 332748118 748683265 1
```

Explanation:

In our sample, for $L = 3$, we have 3 sub-strings :

- aba (Starting from 1st position)
- bab (Starting from 2nd position)
- aba (Starting from 3rd position)

Among them we can chose in the following ways, to get all 3 of them pairwise identical:

- $(1,1,1) : Probability\ = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$
- $(2,2,2) : Probability\ = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$
- $(3,3,3) : Probability\ = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$
- $(1,1,3) : Probability\ = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$
- $(1,3,1) : Probability\ = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$
- $(3,1,1) : Probability\ = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$
- $(3,3,1) : Probability\ = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$
- $(3,1,3) : Probability\ = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$
- $(1,3,3) : Probability\ = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$

$(1,3,3)$ means that the three random substrings chosen are the substring starting at index 1, at index 3 and at index 3.

So $F(3) = 9 \times \frac{1}{27} = \frac{1}{3}$

Nicks Landing (LNDNCK)

In a city called Nick's Landing, there are N people (numbered 1 through N). For each valid i, the i-th person has beauty B_i and politeness P_i; no two people are equally beautiful.

You should answer Q queries. In each query you are given two integers L and R, and you should calculate $Score(L, R)$ as defined below:

```
Score (L, R) :
    Initialize empty Lists Q, OddQ, EvenQ
    For each person with id between [L, R] :
        add his id to List Q
    Sort Q in increasing order of the beauty value of the persons
    Flag = 0;
    For each person's id in Q :
        Flag = 1 - Flag
        If (Flag == 1) append his id to List OddQ
        Else append his id to List EvenQ
    Return DiffSum (OddQ) + DiffSum (EvenQ)

DiffSum (Z[1...n]) :
    Sum = 0
    For i = 2 to n :
        u = Z[i]
        v = Z[i - 1]
        Sum = Sum + |P[u] - P[v]|
    Return Sum
```

Input

- The first line of the input contains a single integer N.
- Each of the next N lines contains two space-separated integers B_i and P_i.
- The next line contains a single integer Q.
- Each of the next Q lines contains two space-separated integers L and R describing a query.

Output

For each query, print a single integer(answer to the query) in a single line.

Constraints

- $1 \leq N, Q \leq 2 \cdot 10^5$
- $1 \leq B_i, P_i \leq 2 \cdot 10^5$
- $1 \leq L \leq R \leq N$
- All beauty values are **distinct**.

Example Input

```
6
2 1
4 6
1 4
3 10
6 4
5 8
5
1 3
1 6
3 6
2 5
4 4
```

Example Output

```
2
15
10
8
0
```

Explanation

For the 2-nd query, $Q = \{1, 2, 3, 4, 5, 6\}$.

If we sort them according to the persons's beauty value, then $Q = \{3, 1, 4, 2, 6, 5\}$.

Now, $OddQ = \{3, 4, 6\}$ and $EvenQ = \{1, 2, 5\}$

So, $DiffSum\ (OddQ) = |P_3$ - $P_4| + |P_4$ - $P_6| = 6 + 2 = 8$.

Similiarly, $DiffSum\ (EvenQ) = |P_1$ - $P_2| + |P_2$ - $P_5| = 5 + 2 = 7$.

So $Score = 7 + 8 = 15$.

Minimize the Distance (MINDIST)

You are given two line segments AB and CD in N-dimensional space. Find the minimum distance between them, i.e. the minimum distance $|PQ|$ over all points $P \in AB$ and $Q \in CD$.

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first line of each test case contains a single integer N.
- The second line contains N space-separated integers $a_1, a_2, \ldots, a_N$ denoting the coordinates of the point A.
- The third line contains N space-separated integers $b_1, b_2, \ldots, b_N$ denoting the coordinates of the point B.
- The fourth line contains N space-separated integers $c_1, c_2, \ldots, c_N$ denoting the coordinates of the point C.
- The fifth line contains N space-separated integers $d_1, d_2, \ldots, d_N$ denoting the coordinates of the point D.

Output

For each query, print a single line containing one real number - the minimum distance between the line segments. Your output will be considered correct if the absolute or relative error does not exceed 10^{-12}.

Constraints

- $1 \le T \le 10^5$
- $3 \le N \le 10^5$
- $0 \le a_i, b_i, c_i, d_i \le 1,000$ for each valid i
- the sum of N over all test cases does not exceed 10^6

Example Input

```
2
3
879 130 645
214 801 973
487 518 681
186 74 684
3
810 702 731
598 325 737
158 871 316
336 999 983
```

Example Output

```
146.899739355990
587.980716360221
```

Prime-partite Graph (PRMPRT)

We have a graph with N nodes (numbered 1 through N). For each valid u and v, there is an undirected edge between nodes u and v if v is a prime divisor of u or u is a prime divisor of v.

You should answer Q queries. In each query, you should find the length of the shortest path between two given nodes a and b or determine that there is no path between them.

Input

- The first line of the input contains a single integer N.
- The second line contains a single integer Q.
- Each of the next Q lines contains two space-separated integers a and b describing a query.

Output

For each query, print a single line containing one integer - the distance between the nodes or -1 if the nodes are not connected.

Constraints

- $1 \leq N, Q \leq 10^6$
- $2 \leq a, b \leq N$

Example Input

```
20
3
2 3
2 6
2 9
```

Example Output

```
2
1
3
```

Explanation

For $a = 2$ and $b = 9$, one possible shortest path is $2 \to 6 \to 3 \to 9$.

Walk 4 steps (WALK4)

You are given a weighted undirected graph consisting of n nodes and m edges. The nodes are numbered from 1 to n. The graph does not contain any multiple edges or self loops.

A walk W on the graph is a sequence of vertices (with repetitions of vertices and edges allowed) such that every adjacent pair of vertices in the sequence is an edge of the graph. We define the cost of a walk W, $Cost(W)$, as the maximum over the weights of the edges along the walk.

You will be given q queries. In each query, you will be given an integer X.
You have to count the number of different walks W of length 4 such that $Cost(W)$ = X.
Two walks are considered different if they do not represent the same edge sequence.

Input:

- First line contains 2 integers : the number of nodes n and number of edges m.
- Next m lines each describe u, v and w, describing an edge between u and v with weight w.
- Next line contains q, the number of queries.
- Next q lines each describe an integer X - the cost of the walk in the query.

Output:

For each query, output in a single line the number of different possible walks.

Constraints

- $1 \leq n \leq 100$
- $1 \leq m \leq \frac{n(n-1)}{2}$
- $1 \leq u, v \leq n$
- $1 \leq w \leq 100$
- $1 \leq q \leq 100$
- $1 \leq X \leq 100$

Sample Input:

```
3 3
1 2 1
2 3 2
3 1 3
3
1
2
3
```

Sample Output:

```
2
10
36
```

EXPLANATION:

For $X = 2$, all possible 10 walks are listed below :

- 1 -> 2 -> 1 -> 2 -> 3
- 1 -> 2 -> 3 -> 2 -> 1
- 1 -> 2 -> 3 -> 2 -> 3
- 2 -> 1 -> 2 -> 3 -> 2
- 2 -> 3 -> 2 -> 1 -> 2
- 2 -> 3 -> 2 -> 3 -> 2
- 3 -> 2 -> 1 -> 2 -> 1
- 3 -> 2 -> 1 -> 2 -> 3
- 3 -> 2 -> 3 -> 2 -> 1
- 3 -> 2 -> 3 -> 2 -> 3

Weighted Necklace (WGTNKLC)

You are given K different colors of beads. A bead of the i-th color has a weight w_i. You have infinitely many beads of each color. All the beads of the same color have the same weight.

You have to answer q independent queries. For each query, you will be given S. You have to form a necklace consisting of n beads such that the total weight of the necklace is S. You can use any number of beads of a single color (possibly zero). Count the number of distinct possible necklaces you can make. Two necklaces will be considered the same if one can be rotated to another.

Input

- First line contains n, denoting the size of the necklace.
- Second line contains K, denoting the number of different colors.
- Third line contains K space separated integers, where the i-th integer w_i denotes the weight of a bead of color i.
- Fourth line contains q, denoting the number of queries.
- Each of the next q lines will contain a single integer S, denoting the total weight of the necklace for this query.

Output

For each query, output in a new line, the number of valid necklaces mod 998244353.

Constraints

- $1 \leq n, q, K, w_i, S \leq 10^5$
- n will be a power of two

Example Input

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

Example Output

```
1
3
5
9
```

Explanation

For S = 6, three ways to form the necklace are:

- 1 -> 1 -> 1 -> 3
- 1 -> 1 -> 2 -> 2
- 1 -> 2 -> 1 -> 2
  Please note, we do not count 2 -> 1 -> 1 -> 2 because it is just a rotation of 1 -> 1 -> 2 -> 2.