National Girls' Programming Contest 2022 (Online Preliminary)

https://toph.co/c/national-girls-programming-contest-2022-preliminary

Schedule

The contest will run for **3h0m0s**.

Authors

The authors of this contest are AK_74u, Anachor, Apurba.884537, Fellow_junior, Iftekhar_Hakim, longlongint, mahdi.hasnat, reborn, shariful_islam, shimul12614, tbs_sajal15, and upobir.

Rules

This contest is formatted as per the official rules of ICPC Regional Programming Contests.

You can use C++11 GCC 7.4, C++14 GCC 8.3, C++17 GCC 9.2, C++20 GCC 12.1, C11 GCC 12.1, C11 GCC 9.2, Java 1.8, PyPy 7.1 (2.7), PyPy 7.1 (3.6), Python 2.7, and Python 3.7 in this contest.

Be fair, be honest. Plagiarism will result in disqualification. Judges' decisions will be final.

Notes

There are 7 challenges in this contest.

Please make sure this booklet contains all of the pages.

If you find any discrepencies between the printed copy and the problem statements in Toph Arena, please rely on the later.

A. Python Interpreter

Sudipa is learning to code in Python 3 and her best friend Sutapa is teaching her a few things everyday. Today Sutapa taught Sudipa about the `print` function.

> If you put anybody's name inside double-quotations and put the whole thing inside parenthesis and put the word `print` before it, when you run the Python program it will print out that name in console.

— said Sutapa

Sudipa was mesmerized after testing it out on her Python console. She wrote

```
print("nazia")
```

and the console printed out

```
nazia
```

She later tried again, this time with her another best friend's name

```
print("eshanee")
```

the console printed out

```
eshanee
```

While Sudipa is absolutely loving to write new codes with print functions that prints out her friends' name, given her code, can you figure out what the console would print out?

Input

The input will contain only one line consisting of a string of characters in the format mentioned above, i.e. in the format `print(“x”)` where **x** will be a non-empty sequence of lower-case English letters only.

The length of the line would be less than 20.

Output

Output what the console would print out if Sudipa's program is run on a Python 3 interpreter.

Samples

Input	Output
`print("sutapa")`	`sutapa`

Input	Output
`print("noshin")`	`noshin`

B. On My Way

One fine morning you discovered yourself standing in a maze. The maze can be considered a **connected weighted undirected** graph. You are currently standing in vertex s and to get out of the maze you need to go to vertex t. But there's a catch. In order to get out of the maze you need to choose a path from s to t such that, the path length differs from the shortest path length of s to t by at-most k. You are wondering how many ways are there to solve the maze.

More formally, count the number of paths from s to t in a connected weighted undirected graph such that, the path length differs from the shortest path length of s to t by at-most k.

As the answer can be huge, you need to print it modulo 998244353.

A path is a sequence of vertices $(v_1, v_2, v_3, \ldots, v_p)$ such that there exists an undirected edge (v_i, v_{i+1}) for $i = 1, 2, \ldots, p - 1$.

Two paths differ, if they contain different number of vertices or there is an index i, where the edge (v_i, v_{i+1}) differs.

A path can visit the same vertex multiple times and even the initial vertex s and final vertex t can be visited multiple times. It is only required that you start at vertex s and finish at vertex t and maintain the path length constraint.

The graph doesn't contain multi-edges and self loops.

Input

The first line of the input contains a single integer $T (1 \le T \le 1000)$— the number of test cases.

The first line of each test case contains three integers n, m and $k (2 \le n \le 2 \times 10^5, n - 1 \le m \le 2 \times 10^5, 1 \le k \le 100)$ — the number of vertices, the number of edges and the path length constraint.

The next line contains two integers s and t $(1 \le s, t \le n, s \ne t)$— the initial vertex and final vertex.

The following m lines describe the edges. Each contains three integers u, v, w $(1 \le u, v \le n, u \ne v, 1 \le w \le 10^9)$ — the undirected edge (u, v) and the weight of the edge.

It is guaranteed that there is at most one edge between any pair of vertices in the graph and the given graph is connected.

The sum of n and the sum of m over all testcases doesn't exceed 2×10^5.

Output

For each testcase, print a single integer — the number of paths modulo 998244353.

Samples

Input	Output
2 6 7 1 1 6 1 2 1 2 3 1 3 4 1 4 5 1 5 6 1 3 5 2 2 4 3 7 10 6 1 7 1 2 1 2 3 1 3 4 1 4 5 1 5 6 1 6 7 1 2 5 2 3 5 2 4 6 3 5 7 4	3 332

Consider the first test case, $s = 1, t = 6$ and $k = 1$.

Shortest path length from s to t is 5. Maximum allowed path length is $5 + 1 = 6$.

Total three paths possible.

$1 \to 2 \to 3 \to 4 \to 5 \to 6$. Length 5.

$1 \to 2 \to 3 \to 5 \to 6$. Length 5.

$1 \to 2 \to 4 \to 5 \to 6$. Length 6.

C. Maximize Sum

You are given two integer arrays A and B of length n and m respectively. You can do the following operations on array A.

- Remove one element from A. This operation can be performed at most k times.
- Insert any element of B into A at any position. You can insert each element of B at most once.

Output the maximum possible value of $\sum_{i=1}^{|A|} A_i \times i$ that you can achieve after performing the aforementioned operations.

Input

Input starts with an integer T, denoting the number of test cases. In each test case,

The first line contains 3 integers n, m, and k.

The second line contains n integers $A_1, A_2, A_3, \ldots, A_n$.

The third line contains m integers $B_1, B_2, B_3, \ldots, B_m$.

Constraints:

- $1 \leq T \leq 10^4$
- $1 \leq n \leq 10^4$
- $1 \leq m, k \leq 10$
- $-10^9 \leq A_i, B_i \leq 10^9$

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

Output

For each test case, print the maximum possible value of $\sum_{i=1}^{|A|} A_i \times i$.

Samples

Input	Output
1 6 5 2 100 -10 20 -2000000 -5 500 -4 -5 -1 -2 70	6147

In the first test case, we shall perform the following operations:

1. From array A, remove -2000000. After removing, array A will become $[100, -10, 20, -5, 500]$.
2. From array B, insert -1 at the beginning of array A. After inserting, array A will become $[-1, 100, -10, 20, -5, 500]$.
3. From array B, insert -2 at the beginning of array A. After inserting, array A will become $[-2, -1, 100, -10, 20, -5, 500]$.
4. From array B, insert -4 at the beginning of array A. After inserting, array A will become $[-4, -2, -1, 100, -10, 20, -5, 500]$.
5. From array B, insert -5 at the beginning of array A. After inserting, array A will become $[-5, -4, -2, -1, 100, -10, 20, -5, 500]$.
6. From array B, insert 70 at the 8th position of array A. After inserting, array A will become $[-5, -4, -2, -1, 100, -10, 20, -5, 70, 500]$.

After completing those operations sequentially, array A gives the maximum result.

$-5 \times 1 + -4 \times 2 + -2 \times 3 + -1 \times 4 + 100 \times 5 + -10 \times 6 + 20 \times 7 + -5 \times 8 + 70 \times 9 + 500 \times 10 = 6147$

Input	Output
5 4 1 1 -160 -290 -693 -597 38 2 2 3 -149 990 46 48 4 6 2 475 -164 -170 456 59 11 26 -45 -76 -65 6 4 4 997 -711 871 251 617 -306 84 -84 -20 -31 1 5 1	-2379 4047 7160 17034 1198

Input	Output
298 -98 -9 -1 -95 -87	

D. Greedy Grid Game

“Uban game world” is an indoor gaming zone. It offers 5 game segments for only one ticket. Today, Aliban is going to participate in one of the games called “Greedy Grid Game”.

This game is about an $n \times m$ grid. Each cell contains an integer value $a_{i,j}$. Here $a_{i,j}$ denotes the number in the i^{th} row from the top and j^{th} column from the left. Aliban has to start the journey from $(1,1)$ and finish at (n,m). She can't move any cell as she pleases. There are some rules that follows -

1. If she starts her movement by going right, i.e., (i,j) to $(i,j+1)$, then she is not allowed to stop until she reaches the border cell (i,m) of the grid.
2. If she starts her movement by going down, i.e., (i,j) to $(i+1,j)$, then she is not allowed to stop until she reaches the border (n,j) cell of the grid.
3. She is allowed to give **at most one** diagonal move in her whole journey. For example, (i,j) to $(i+1,j+1)$, (i,j) to $(i+1,j-1)$, (i,j) to $(i-1,j+1)$, (i,j) to $(i-1,j-1)$.
4. She is not allowed to visit any cell more than once.
5. Except for the diagonal moves, she cannot move left or up.
6. She can not move outside of the grid.

The score of Aliban will be the sum of cell values she visits throughout her journey.

As Aliban is not an expert in calculation she asks for your help. You have to find the maximum score she can achieve without violating any rules.

Input

The input consists of multiple test cases. The first line contains an integer $t (1 \le t \le 100)$ — the number of test cases. The description of the t test cases follows.

The first line of each test case contains two positive integers n and $m (1 \le n, m \le 10^5; 1 \le nm \le 10^5)$— the number of rows and columns respectively.

The following n lines contain m integers each, the j^{th} element in the i^{th} line holds the value $a_{i,j}$ $(-10^6 \le a_{i,j} \le 10^6)$.

It is guaranteed that the sum of nm over all test cases does not exceed 10^5.

Output

For each test case, print the maximum score Aliban can achieve.

Samples

Input	Output
2 3 3 1 2 2 1 10 -2 2 -3 1 3 3 -1 2 3 -4 5 6 10 2 3	14 19
In the first test case, to achieve the best score her movement should be $(1,1) \to (1,2) \to (1,3) \to (2,2) \to (2,3) \to (3,3)$	

E. GCD, Divisor, Count!

I like short statements
and you should too
so here's the statement
without further ado

You are given 3 positive integers N, A, B. You have to count the number of N length positive integer arrays such that each of the elements of the array are divisors of B and the greatest common divisor of the elements is A.
Since the count can be large you have to output it modulo 1000000007. In other words, if the count is X, you have to output $X \bmod 1000000007$.

Input

There will be T independent testcases. First line of input will contain the positive integer T. Next T lines will each contain three positive integers N, A, B separated by spaces.

$1 \leq T \leq 100$
$1 \leq N \leq 10^9$
$1 \leq A, B \leq 10^{12}$

Output

Output T lines. The i th line should have the answer for the i th testcase.

Samples

Input	Output
3 2 2 16 1 5 15 3 6 15	7 1 0

In first testcase, there are 7 such arrays $[2, 2], [2, 4], [2, 8], [2, 16], [4, 2], [8, 2], [16, 2]$

F. Cardboard Mountains

Alice has been given the responsibility of creating the background of the annual play in her school. The background consists of Mountains. One such Background-

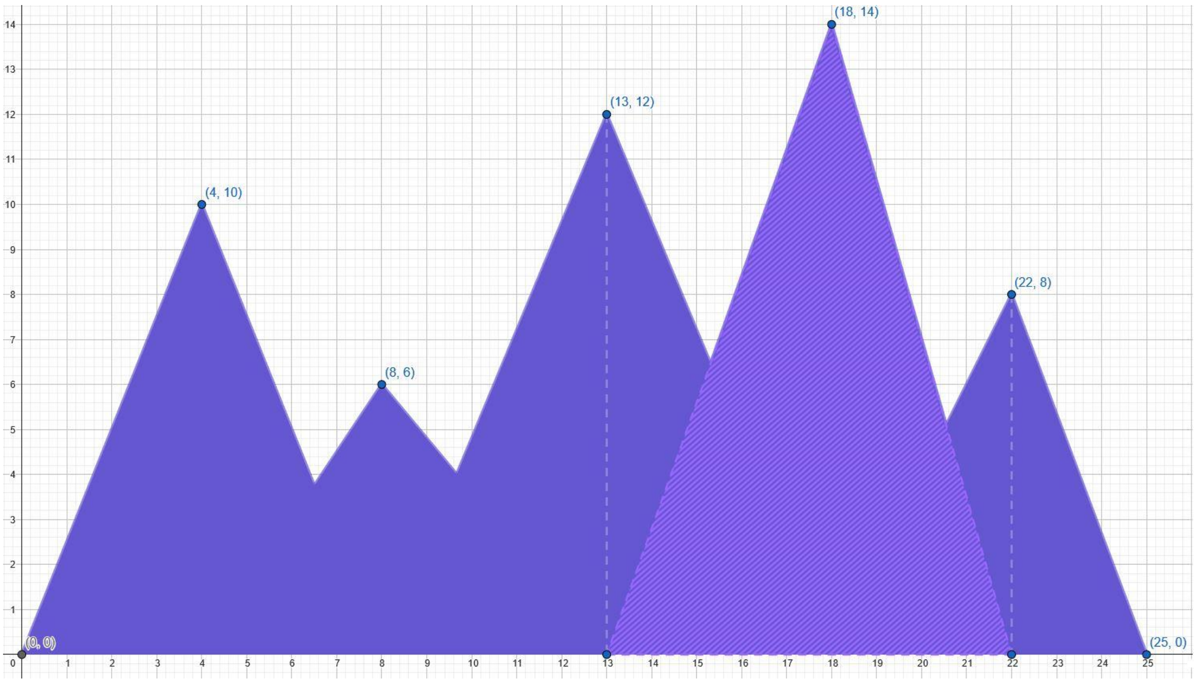

Figure 1: A Cardboard Mountain Background.

The Mountains' characteristics are —

1. Mountains are triangles whose base is on the floor.
2. A stick is placed vertically from the floor to the peak of the mountain to support it.
3. The lower right vertex of a mountain starts from the stick of the mountain to the immediate left and the lower right vertex on the stick of the mountain to the immediate right.
4. The left-most mountain's lower left vertex is on the lower left corner of the background.
5. The right-most mountain's lower right vertex is on the lower right corner of the background.
6. The width of the stick is negligible.

Note that, a mountain m_1 is defined left (or, right) to another mountain m_2 if the stick supporting m_1 is strictly on the left (or, right respectively) of m_2. Furthermore, m_1 is the immediate left (or, right) of m_2 if among all of the mountains who are on left (or, right) of m_2, m_1 is the right-most (or, left-most respectively).

The Mountains are traditionally created this way. But Alice being a problem solver has thought of an easier way to create the mountains. Alice thought of doing the following —

1. Consider the background as a 2D Cartesian plane, where the lower left corner will be the origin and the lower right corner will be $(X, 0)$, where X is the length of the background.
2. Fix the coordinates of the peaks of N Mountains on the Cartesian plane, then draw the Mountains respecting their characteristics, on a paper for reference.
3. Cut a single cardboard shape according to that reference.

Let's say Alice has generated 5 mountain peaks' co-ordinates. They are $(8, 6)$, $(22, 8)$, $(4, 10)$, $(13, 12)$ and $(18, 14)$. The length of the background is 25 units. The final shape of the cardboard shape can be denoted by *figure 1*.

Alice is wondering what would be the area of the final cardboard shape for the Mountains on the background after cutting. Given the coordinates of the N mountain tips, and the length of the background, X, can you calculate the area of the final cardboard shape?

Input

The first line contains a single integer T $(1 \leq T \leq 10^4)$, the number of test cases.

The first line of each test case contains two space-separated integers, $N(1 \leq N \leq 2 \times 10^5)$ and $X(2 \leq X \leq 10^6)$, denoting the number of peaks and the background's length respectively.

The i^{th} line of the next Nlines contain two integers x_i and y_i $(0 < x_i < X,\ 1 \leq y_i \leq 10^6)$, the coordinate of the i^{th} mountain.

The sum of N all over test cases doesn't exceed 2×10^5.

Output

For each test case, output the area of the final cardboard shape. Your answer will be considered correct if the relative or absolute difference is less than 10^{-6}. That means, if your answer is A and jury's answer is B, your answer will be considered correct if $\frac{|A-B|}{max(1,B)} < 10^{-6}$.

Samples

Input	Output
1 5 25 8 6 18 14 4 10 13 12 22 8	174.16433566

G. Blanket

You have a suitcase with length X, width Y and height Z. You also have a very comfortable *blanket*. It has length L, width W and thickness H. You love your *blanket* very much. You like to take it with you, wherever you go. To fit the *blanket* in your suitcase, you can apply a *fold* operation on the *blanket*.

One *fold* operation is as follows:

- Divide the length or width by 2.
- Multiply the thickness by 2.

The *blanket* fits if you can keep it in the suitcase in an axis aligned way. Formally, at least one corner of the *blanket* must exactly coincide with a corner of the suitcase keeping the entire *blanket* inside.

Your task is to find the minimum number of *fold* operations required to fit the *blanket* in the suitcase. Or report if it is impossible.

Note that it is not necessary to keep the suitcase in input orientation of X * Y * Z, as long as the orientation is axis aligned.

Input

The first line contains an integer $T (1 \leq T \leq 10^5)$ the number of test cases.
Each of the next T lines describe a test case with 6 space separated integers X, Y, Z, L, W, H. Each of these 6 integers are greater than 0 and do not exceed 10^9.

Output

For each test case print a single integer in a separate line.
Print the minimum number of *fold* required to fit the *blanket*. If impossible, print -1.

Samples

Input	Output
4 100 100 100 1600 100 6 100 100 100 1600 100 7 32 1024 2048 4096 16384 1 1000000000 10000 100 98 500000000 9899	4 -1 5 0

In the first case, you can *fold* the length 4 times to obtain $(L, W, H) = (100, 100, 96)$.
In the second case, no sequence of operation can make it possible.
In the third case, after *fold*-ing the length and width 1 time and 4 times respectively, you can obtain $(L, W, H) = (2048, 1024, 32)$. Which fits the suitcase exactly. **A suitcase $(32, 1024, 2048)$ can be oriented in an axis aligned way to $(2048, 1024, 32)$.**
In the fourth case, it already fits. So no operation is necessary.