OUTPUT QUESTIONS

Q1

How many ways are there to place a black and a white knight on an **N \* M** chessboard such that they do not attack each other? The knights have to be placed on different squares. A knight can move two squares horizontally and one square vertically (L shaped), or two squares vertically and one square horizontally (L shaped). The knights attack each other if one can reach the other in one move.

**Input:**  
The first line contains the number of test cases T. Each of the next T lines contains two integers N and M which is size of matrix.

**Output:**  
For each testcase, print the required answer, i.e, number of possible ways to place knights.

**Constraints:**  
1 <= T <= 100  
1 <= N, M <= 105

**Example:**  
**Input:**  
3  
2 2  
2 3  
4 5

**Output:**  
12  
26  
312

Q2

Jeny love Sweets so much. Now she is at famous restaurant and wants to eat **M** pieces of a particular sweet. Cost of **nth** sweet can only be determined by the determinant of matrix of order **n x n,**where n = 1 to M. The (i, j)th term of matrix is given as:  
  
A[i][j]= minimum(i, j) \*(-1)((i-1)\*n + (j-1)).  
  
Matrix indexes starts with 1. The task is to find the cost of M sweets.

**Input:**The First line of input contains a integer T, denoting the total number of test cases. Then T test cases follow. Each test case consists of a single line conataining an integer M denoting the number of Sweets.

**Output:**Print the desired output corresponding to each test case in a separate line.  
  
**Constraints:**  
1<=T<=100  
1<=M<=1000000

**Example:**  
**INPUT :**  
2  
1  
2

**OUTPUT:**  
1  
0

Q3

Given an incomplete Sudoku configuration in terms of a 9x9  2-D square matrix (mat[][]) the task to check if the configuration has a solution or not.   
  
**Input:**  
The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains 9\*9 space separated values of the matrix mat[][] representing an incomplete Sudoku state where a 0 represents empty block.  
  
**Output:**  
For each test case in a new line print 1 if the sudoku configuration is valid else print 0.  
  
**Constraints:**  
1<=T<=10  
0<=mat[]<=9  
  
**Example:**  
**Input:**  
2  
3 0 6 5 0 8 4 0 0 5 2 0 0 0 0 0 0 0 0 8 7 0 0 0 0 3 1 0 0 3 0 1 0 0 8 0 9 0 0 8 6 3 0 0 5 0 5 0 0 9 0 6 0 0 1 3 0 0 0 0 2 5 0 0 0 0 0 0 0 0 7 4 0 0 5 2 0 6 3 0 0  
3 6 7 5 3 5 6 2 9 1 2 7 0 9 3 6 0 6 2 6 1 8 7 9 2 0 2 3 7 5 9 2 2 8 9 7 3 6 1 2 9 3 1 9 4 7 8 4 5 0 3 6 1 0 6 3 2 0 6 1 5 5 4 7 6 5 6 9 3 7 4 5 2 5 4 7 4 4 3 0 7   
  
**Output:**  
1  
0

Q4

Given a set of **N** nuts of different sizes and N bolts of different sizes. There is a one-one mapping between nuts and bolts. Match nuts and bolts efficiently.

Comparison of a nut to another nut or a bolt to another bolt is not allowed. It means nut can only be compared with bolt and bolt can only be compared with nut to see which one is bigger/smaller.

**Input:**  
The first line contains '**T**' denoting the number of testcases. Then follows description of T testcases:  
Each case begins with a single positive integer N denoting the number of nuts/bolts. Then follows the array of nuts, each element separated by a space. And finally the bolts array, again, each element is separated by a space here. Array of Nuts/Bolts can only consist of the following elements:{'@', '#', '$', '%', '^', '&', '~', '\*', '!'}. And no element can be repeated.

**Output:**  
For each test case, output the matched array of nuts and bolts in separate lines, where each element in the array is separated by a space. Print the elements in the following order ! # $ % & \* @ ^ ~

**Constraints:**  
1 <= T <= 70  
1 <= N <= 9

**Example:**  
**Input:**  
2  
5  
@ % $ # ^  
% @ # $ ^  
9  
^ & % @ # \* $ ~ !  
~ # @ % & \* $ ^ !

**Output:**  
# $ % @ ^  
# $ % @ ^  
! # $ % & \* @ ^ ~  
! # $ % & \* @ ^ ~

Q5

Consider a big party where a log register for guest’s entry and exit times is maintained. Find the time at which there are maximum guests in the party. Note that entries in register are not in any order.  
  
**Input:**  
The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. Each test case contains an integer n denoting the size of the entry and exit array. Then the next two line contains the entry and exit array respectively.  
  
**Output:**  
Print the maximum no of guests and the time at which there are maximum guests in the party.

**Constraints:**  
1<=T<=10^5  
1<=N<=10^5  
1<=entry[i],exit[i]<=10^5

**Example:**  
**Input:**  
2  
5  
1 2 10 5 5  
4 5 12 9 12  
7  
13 28 29 14 40 17 3   
107 95 111 105 70 127 74   
  
**Output:**  
3 5  
7 40

Q6

Given an unsorted array of size **N** of positive integers. **One number 'A'** from set {1, 2, …N} is missing and **one number 'B'** occurs twice in array. Find these two numbers.

**Note**: If you find multiple answers then print the Smallest number found. Also, expected solution is O(n) time and constant extra space.

**Input:**  
The first line of input contains an 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 denoting the size of array. The second line contains N space-separated integers A1, A2, ..., AN denoting the elements of the array.

**Output:**  
Print B, the repeating number followed by A which is missing in a single line.

**Constraints:**  
1 ≤ T ≤ 100  
1 ≤ N ≤ 106  
1 ≤ A[i] ≤ N

**Example:**  
**Input**:  
2  
2  
2 2  
3   
1 3 3

**Output**:  
2 1  
3 2

Q7

Pasha has been very sick. His platelets went way down. Being a paranoid person, he consulted N doctors about the optimal range in which Platelet Count should lie. The i-th doctor suggested that the Platelet count should be between li and ri, inclusive, to be called normal.  
Now, Pasha thinks that a Platelet count is Safe to have if at least Z Doctors recommend it. Pasha now asks Q Queries. In each query- he will give an integer P (the platelet count). Pasha wants to know if the entered Platelet count is safe to have or not.

**Input:**  
The first line contains T-denoting the number of test cases.  
The first line of each test case contains an integer N -(Number of Doctors).  
The next N Lines each contains two integers li and ri.  
The next Line contains two integers -  Z(Number of doctors that should recommend that platelets count is safe; at least) and Q(Number of queries being asked)  
The next Q lines each contains an integer P.   
**Output:**  
For each test case-Output -"Yes", if its safe to have, else "No".

**Constraints:**  
**1<=T<=100**  
**1<=N,Q<=2e5**  
**1<=Z<=N**  
**1<=Li<=Ri<=2e5**  
**1<=P<=2e5**

**Explanation:**  
**Sample Input:**  
1  
5  
1 10  
5 10  
7 12  
15 25  
20 25  
3 4  
7  
5  
10  
16

**Sample Output:**  
Yes  
No  
Yes  
No

Q8

Given a unsorted array, you need to find the maximum difference of absolute values of elements and indexes, i.e., for **i < j**, calculate maximum of **| A[i] - A[j] | + | i - j |.**

**Input:**The first line of input contains an integer **T** denoting the number of test cases. Then **T** test cases follow. Each test case consists of two lines. The first line of each test case contains **N** where N is the size of array. The second line of each test case contains **N** space separated integers denoting the array elements.

**Output:** For each testcase in the new line, print the maximum diiference as given in question.

**Constraints:**  
1 <= **T** <= 100  
1 <= **N** <= 106  
0 <= **|A[i]|** <= 106

**Example:**  
**Input:**  
1  
3  
1 3 -1  
**Output:**  
5

Q9

Given an array containing N numbers. The task is to find the maximum sum bitonic subarray. A bitonic subarray is a subarray in which elements are first increasing and then decreasing. A strictly increasing or strictly decreasing subarray is also considered as bitonic subarray.

**Time Complexity** : O(n)  
**Auxiliary Space** : O(1)

**Input:**  
The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. Each test case consists of two lines. First line of each test case contains a integer N and the second line contains N space separated array elements.

**Output:**  
For each test case, print the maximum bitonic subarray sum in new line.

**Constraints:**  
1<=T<=200  
1<=N<=105  
1<=A[i]<=105

**Example:**  
**Input:**  
2  
7  
5 3 9 2 7 6 4  
7  
5 4 3 2 1 10 6

**Output:**  
19  
17

Q10

Indian Air Force fighter planes are located in different bases across the country. Each airbase is described by some integer coordinate (x,y)(x,y). The Air Force plans to do surgical strikes on a maximum of MM different targets in enemy territory (which are also described by cartesian coordinates) and then come back to the common main airbase at coordinate (baseX,baseY)(baseX,baseY) .

Each army base and the targets are recognised by a secret integer IDID. The time taken for an aircraft to go from a base to a target is the prime factor of the Manhattan Distance between the base and the target that is just greater than the IDID of the source base (In case the IDID is greater than or equal to the largest prime factor, then consider the IDID itself). Similarly, the time taken for an aircraft to go from a target to the main base is the prime factor of the Manhattan Distance between the target and the main base that is just greater than the IDID of the target (In case the IDID is greater than or equal to the largest prime factor, then consider the IDID itself).

Each Aircraft needs to leave the base, reach target and come back to the main base in a maximum time of TT. One aircraft can go to only one target before going to the main base.

Find the maximum number of targets that can be reached in the enemy territory.

Input

* The first line contains three space separated integers NN, MM and TT respectively.
* The next NN lines contain 3 integers denoting xx coordinate, yy coordinate and the IDID of the air bases.
* The next MM lines contain 3 integers denoting xx coordinate, yy coordinate and the IDID of the targets.
* The last line contains two integers denoting the baseXbaseX and baseYbaseY coordinate.

Output

Output a single integer which is the maximum number of targets that can be reached.

Constraints

* 1≤N,M≤4001≤N,M≤400
* 0≤x,y,baseX,baseY≤5∗1060≤x,y,baseX,baseY≤5∗106
* 0≤ID≤50000≤ID≤5000
* 0≤T≤1070≤T≤107

Sample Input

2 2 35  
1 2 15  
2 10 20  
2 1 8  
5 6 12  
5 5

Sample Output

2

Q11

Given an array of size NN and two integers KK and SS, the special sum of a subarray is defined as follows:  
(Sum of all elements of the subarray) \* (KK - pp \* SS)  
Where pp = number of distinct prime factors of “product of all elements of the subarray”.  
Find the maximum special sum by considering all non-empty subarrays of the given array.

Input

* First line contains 3 integers NN, KK and SS.
* Second line contains NN integers, the elements of the array.

Output

Output a single integer. The maximum special sum considering all non-empty subarrays of the array.

Constraints:

* 1≤N,K,S≤1051≤N,K,S≤105
* 0≤K/S≤200≤K/S≤20
* 1<1< Any element of array <105<105

Sample Input

4 10 2  
14 2 7 15

Sample Output

138

Q12

You are given a sequence of NN powers of an integer kk; let's denote the ii-th of these powers by kAikAi. You should partition this sequence into two non-empty contiguous subsequences; each element of the original sequence should appear in exactly one of these subsequences. In addition, the product of (the sum of elements of the left subsequence) and (the sum of elements of the right subsequence) should be maximum possible.

Find the smallest position at which you should split this sequence in such a way that this product is maximized.

Input

* The first line of the input contains a single integer TT denoting the number of test cases. The description of TT test cases follows.
* The first line of each test case contains two space-separated integers NN and kk.
* The second line contains NN space-separated integers A1,A2,…,ANA1,A2,…,AN.

Output

For each test case, print a single line containing one integer — the size of the left subsequence. If there is more than one possible answer, print the smallest possible one.

Constraints

* 1≤T≤101≤T≤10
* 2≤N≤1052≤N≤105
* 2≤k≤1092≤k≤109
* 0≤Ai≤1050≤Ai≤105 for each valid ii

Subtasks

Subtask #1 (30 points):

* 2≤N≤1,0002≤N≤1,000
* 0≤Ai≤1,0000≤Ai≤1,000 for each valid ii

Subtask #2 (70 points): original constraints

Example Input

1

5 2

1 1 3 3 5

Example Output

4

Q13

For a permutation P = (p1, p2, ..., pN) of numbers [1, 2, ..., N], we define the function f(P) = max(p1, p2) + max(p2, p3) + ... + max(pN-1, pN).

You are given N and an integer K. Find and report a permutation P of [1, 2, ..., N] such that f(P) = K, if such a permutation exists.

Note f([1]) = 0.

Input

* The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows.
* The only line of each test case consists of two space-separated integers N, K respectively.

Output

For each test case, if a permutation satisfying the condition exists, output a single line containing N space-separated integers which denotes any such permutation. If no such permutation exists, output a single integer -1 instead.

Use fast I/O methods since the size of the output is large.

Constraints

* 1 ≤ T ≤ 40
* 1 ≤ N ≤ 105
* Sum of N over all test cases in each file ≤ 106
* 0 ≤ K ≤ 2 \* 1010

Example

Input:

3

4 12

2 2

5 14

Output:

-1

1 2

5 4 3 2 1

Explanation

Example 1. There doesn't exist any permutation of numbers [1, 2, 3, 4] that can have its f value equal to 4. Hence answer is -1.

Example 2. The permutations [1, 2] and [2, 1] both have their f values equal to 2. You can print any of these two permutations.

Example 3. The permutation [5, 4, 3, 2, 1] has f value = max(5, 4) + max(4, 3) + max(3, 2) + max(2, 1) = 5 + 4 + 3 + 2 = 14.

Q14

There is a universal library, where there is a big waiting room with seating capacity for maximum mm people, each of whom completes reading nn books sequentially. Reading each book requires one unit of time.

Unfortunately, reading service is provided sequentially. After all of the mm people enter the library, the entrance gate is closed. There is only one reading table. So when someone reads, others have to wait in the waiting room.

At first everybody chooses nn books they want to read. It takes xx amount of time. People can choose books simultaneously. Then they enter the waiting room. After reading nn books the person leaves the library immediately.

As nothing is free, the cost of reading is also not free. If a person stays in the library tt units of time then the cost of reading is ⌊t−nm⌋⌊t−nm⌋ units of money. So, the ithith person pays for time xx he needs to choose books and the time (i−1)∗n(i−1)∗n he needs to wait for all the persons before him to complete reading.

Note: ⌊a⌋⌊a⌋ denotes the floor(aa).

Input

* Each case contains three space-separated positive integers nn, mm and xx where n,x≤1000n,x≤1000 and m≤1015m≤1015.
* End of input is determined by three zeros.
* There are no more than 1000 test cases.

Output

* For each case, output in a single line the total unit of money the library gets in that day.

Sample Input

1 100 9

11 2 10

12 2 11

0 0 0

Sample Output

9

15

16

Q15

In a Circular City, there are nn houses, numbered from 1 to n and arranged in 1,2,…,n,1,2,… Chef needs to deliver packages to mm (m<=n) houses.

Chef is initially at house 1. Chef decides an integer xx and stops after every xx houses. i.e- if n=7n=7 and x=2x=2. He will stop at 1,3,5,7,2,… He may deliver a package when he stops at a house. His work is done when all the packages are delivered.

What is the minimum number of times Chef has to stop, if he can choose any xx ?

Note: Starting point (1) is also counted in number of stops

Input:

* First line will contain n,mn,m, denoting number of houses and number of packages respectively.
* Next line contains mm distinct space separated integers denoting the houses

Output:

Single line containing an integer denoting minimum number of stops.

Constraints

* 3≤n≤10003≤n≤1000
* 1≤m≤n1≤m≤n

Sample Input 1:

5 3

1 2 4

Sample Output 1:

3

Sample Input 2:

6 2

3 4

Sample Output 2:

4

Q16

Chef is an advocate for Go Green Initiative. Today he had n trees planted in a row outside his his restaurant. Today, the height of i-th tree is hi feet. The trees grow at a rate of mi feet per day.

Chef knows that trees will look beautiful if they form a *zig-zag* sequence. The trees will be said to be in *Zig-zag* sequence if the heights of tree first increases or decreases, then alternates between decreasing/increasing respectively. Formally, the trees will be said to in *Zig-zag* sequence if one of the following two conditions holds.

* h1 < h2 > h3 < h4 and so on..
* h1 > h2 < h3 > h4 and so on..

Chef wants to know intervals of time when the heights of the trees will form a *zig-zag* sequence.

Input

The first line of the input contains an 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, denoting the number of trees.

The ith of following N lines contains two space separated integers hi and mi, denoting the initial height and the growth speed for ith tree.

Output

For each test case, output an integer Q - the amount of the periods of consecutive moments of time, when the trees for a zig-zag sequence.

On the following Q lines, output the intervals of time when the trees' heights form a zig-zag sequence. For each intervals, output its' smallest and the largest instants of time. If the range is infinite, output Inf as the right bound.

The test cases are designed in such a way that the total output won't exceed 2 MB.

Constraints

* 1 ≤ T ≤ 105
* 1 ≤ n ≤ 10
* Subtask 1 (23 points): 0 ≤ hi, mi ≤ 10
* Subtask 2 (77 points): 0 ≤ hi, mi ≤ 109
* 1 ≤ sum of n over a test cases in a single test file ≤ 5 × 105

Example

Input:

3

3

0 1

2 2

0 3

2

2 1

1 2

3

1 1

2 2

3 3

Output:

1

0 1

2

0 0

2 Inf

0

Q17

 recent glut in Chefland's markets has caused the local currency *Peppercorn* to devaluate sharply. (Peppercorns are getting cheaper on an average, though there could be ups and downs).

And Chef needs to rescue his wealth! Initially, he had D Peppercorns. There are N exchange kiosks in the city he lives in, where he can go and exchange his Peppercorns for a well-known stable currency, the Antarctican Dollar. For each kiosk, the exchange rate for the first M seconds of the day is known (both for buying and selling). All kiosks are arranged in a row, and to travel from the ith to the jth kiosk, you need to spend |i-j| seconds, and to exchange currency at any kiosk, you also need 1 second. So, starting at point of time 0 at any of the kiosks (he can get there before trading starts for the day), Chef runs from one kiosk to another to buy and sell currency. You need to find the maximum amount of Peppercorns Chef could have after the Mth second.

Note

* If X is a buying rate, the kiosk will pay you X Peppercorns for 1 Antarctican Dollar.
* If X is a selling rate, you will pay the kiosk X Peppercorns for 1 Antarctican Dollar.

Input

First line of input contains three numbers — N, M and D. N lines follow. ith line (i = 0 … N-1) contains 2\*M integers — currency rate for ith kiosk. Numbers Ai, 2j and Ai, 2j+1 (j = 0 … M-1) are the selling and buying rates, respectively, at the jth second for the ith kiosk.

Output

Output a single number: the maximum amount of money (in Peppercorns - in the end Chef ought to have all the money converted to local currency since it's the only valid currency in the country for financial operations) he could have after M seconds, with absolute or relative error not more than 1e-6.

If the amount of money of any currency that Chef will own at any point of time exceeds 1018, output file should contain only a single line containing the string *“Quintillionnaire”* (without quotes, followed by a newline character).

Constraints

* 1 ≤ D ≤ 1018
* 1 ≤ N, M ≤ 103
* 1 ≤ Ai, j ≤ 109
* Ai, 2j > Ai, 2j+1 (because there are no miracles in Chefland — you’ll always pay more than what the kiosk will pay you. Otherwise, one could’ve indefinitely sold and bought and increased their money this way).

Subtasks

Subtask 1 (20 points):

* 1 ≤ N ≤ 100
* 1 ≤ M ≤ 100
* 1 ≤ D ≤ 103
* 1 ≤ Ai,j ≤ 103

Subtask 2 (30 points):

* 1 ≤ N ≤ 100
* 1 ≤ M ≤ 100

Subtask 3 (50 points): No additional constraints

Example

Input:

3 3 5

2 1 5 3 7 6

2 1 4 3 6 5

10 9 8 7 6 5

Output:

15.0000000000

Q18

Your teacher is very evil and gives you a lot of homework. In fact, the homework she gives you takes so much time that you get barely anytime to sleep. Today, she gave an extra-long homework: The teacher has given you a list A of N distinct integers, Q queries and an integer M. Each query will have two integers, i and R. You have to find j, such that a[j] is the ith smallest element for which a[j]%M = R. .

### Input

The first line of each test case contains three integer N, Q and M denoting the number of elements in the array, number of queries and the modulus respectively The second line contains N space-separated integers A1, A2, ..., AN denoting the elements in the array. The next Q contains two integers: i and R.

### Output

For each query output one integer, j(1-indexed), which is the index such that a[j] is the ith smallest element for which a[j]%M = R. If it doesn't exist then print -1

### Constraints

 1 ≤ N, Q, M ≤ 2\*105

 0 ≤ R ≤ M

 1 ≤ A[i] ≤ 2\*108

### Subtasks

 Subtask 1(20 points): N\*Q ≤ 106,

 Subtask 2(80 points): Original constraints

### Example

Input:

8 5 4

1 5 7 3 9 11 0 4

1 1

2 1

3 1

1 3

5 3

Output:

1

2

5

4

-1

Q19

Mark has recently started studying string algorithms. So, as to gauge his knowledge, Lucy challenges him to a task.

"Given a string s, answer several times a query to determine whether a *substring* s[i, j] (inclusive) is palindromic or not.", said Lucy in a confident tone!

As smart as Mark is, he was able to instantly find the solution!

Now, Mark has challenged little Lucy to do the same task by reversing a specific substring beforehand. As Lucy is still just a novice, she asks for your help.

You have to write a program that answers Q queries on a string S.

Each query contains four integers (i, j, k, l). For every query, first reverse the *substring* s[i, j] (inclusive) and then report if *substring* s[k, l] (inclusive) is a palindrome.

Note that the reversal operations are only for the specific query and should not persist for further queries. Please check the explanation section for better understanding.

Input:

The first line of input file contains string S. The next line contains an integer Q.  
Each of the following Q lines each contain 4 space separated integers i, j, k and l.

Output:

Output exactly *Q* lines, each containing the result of corresponding query as *"Yes"* or *"No"*.

Constraints:

* 1 ≤ |S| ≤ 105
* 1 ≤ Q ≤ 333333
* 1 ≤ i ≤ j ≤ |S|
* 1 ≤ k ≤ l ≤ |S|
* S contains only the characters 'a' to 'z'

Example:

Sample Input:

ababa

4

2 3 3 4

1 2 3 4

1 3 3 5

2 4 1 5

Sample Output:

Yes

No

Yes

Yes