Problem A – Cool number

A positive integer X is a cool number if it satisfies the following conditions:

- It has k digits $(1 \leq k \leq 10)$, numbered from 1 to k, with the 1st digit being the most significant and the kth digit being the least significant.
- The value of the i-th digit equals to the number of occurrence of digit $i - 1$ in X.

For example:

- 21200 is a cool number because it has 2 digits 0, 1 digit 1, 2 digits 2, 0 digit 3, and 0 digit 4.
- 8000000010 is not a cool number because the second digit is 0 but there is one digit 1.

Input

The first line contains an integer T $(1 \leq T \leq 10)$ denoting the number of tests. Then T tests follow, each test case consists of only one number K.

Output

For each test case, print out all the cool numbers in one line, separated by a single space. If there is no cool number, output "-1"

Sample

Sample input	Output for sample input
2 1 5	-1 21200

Problem B – K-smallest numbers

Given an array A of N integers $A_1, A_2, A_3, \dots, A_N$. This array is generated by the following formula:

$$A_1 = S$$
$$A_i = (C_1 \times A_{i-1} + C_2) \ modulo \ M, \qquad i = 2 \to N$$

Your task is very simple. Find out K smallest integers among the N generated integers.

Input

First line of input consist number of test case T ($T \leq 20$). Each of next T line describes a test case. Each test case consists of six numbers: N, K, S, C_1, C_2, M where

- $1 \leq N \leq 10^7$
- $K \leq \min\ (N, 100)$
- $0 \leq S, C_1, C_2 \leq 10^7$
- $2 \leq M \leq 10^7$

Output

For the i-th test case, you should print K smallest numbers in ascending order on the i-th line of output. The numbers should be separated by a single space.

Sample

Sample input	Output for sample input
2 10000 9 71 197 5 91723 2014 4 27 279 64 97264	3 4 4 24 40 40 41 49 71 27 77 221 251

Problem C – Grid city

Alpha is a modern and very well-planning city. The city is arranged in a grid shape with all the streets are one-way and parallel to the North-South axis or East-West axis. There are V vertical streets (parallel to North-South axis) and H horizontal streets (parallel to East-West axis). V vertical streets are numbered 1 to V from West to East. H horizontal streets are numbered 1 to H from South to North.

If we represent this city in a 2D plane, the first vertical street is on the line $x = 0$, the first horizontal street is on the line $y = 0$. The i-th vertical street is VG_i meters from the $(i + 1)$-th vertical street. The j-th horizontal street is HG_j meters from the $(j + 1)$-th horizontal street.

Vertical streets are either Northbound (go from South to North) or Southbound (go from North to South). The directions of these vertical streets are given in a string VD where VD_i is either 'N' or 'S'. Horizontal streets are either Westbound (go from East to West) or Eastbound (go from West to East). The directions of these horizontal streets are given in a string HD where HD_j is either 'W' or 'E'.

Given K queries x_1, y_1, x_2, y_2, you are to calculate the shortest path from (x_1, y_1) to (x_2, y_2).

Input

The first line is the number T ($T \leq 20$) denotes the number of test cases. Then T test cases follow:

- The first line of each test is 3 integers V, H, K. ($1 \leq V, H \leq 5000, 1 \leq K \leq 1000$)
- The second line of each test is VG - an array of length $V - 1$. ($1 \leq VG_i \leq 1000$)
- The third line of each test is HG - an array of length $H - 1$. ($1 \leq HG_j \leq 1000$)
- The fourth line of each test is VD - a string of length V.
- The fifth line of each test is HD - a string of length H.
- Then K queries follow. Each query consists of 4 non-negative integers x_1, y_1, x_2, y_2. It is guarantee that each points lies on a street.

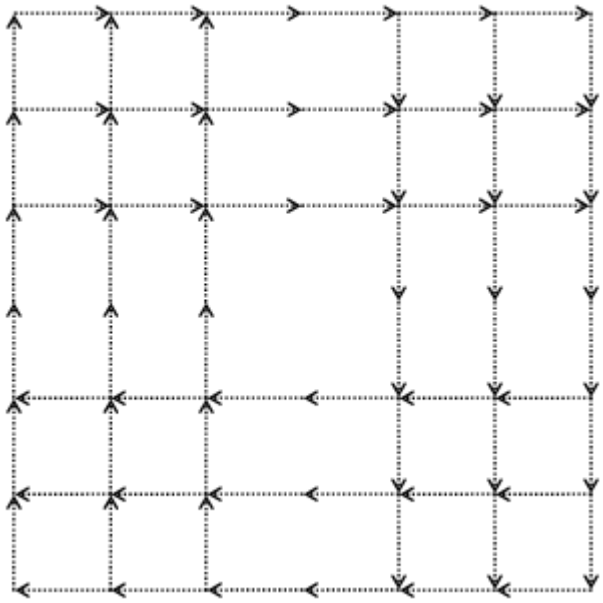
Output

For each query, print the shortest distance. If it is not possible to go from (x_1, y_1) to (x_2, y_2), print -1 instead.

Sample

Sample input	Output for sample input
1 6 6 4 1 1 2 1 1 1 1 2 1 1 NNNSSS WWWEEE 5 4 5 4 2 2 4 4 3 2 3 0 6 6 5 5	0 4 10 14

The city map with directions:



Problem D – Pairwise coprime set

Two positive numbers are called coprime (or relatively prime) if their greatest common divisor is 1. A set of integers is said to be pairwise coprime if a and b are co-prime for every pair (a, b) of different integers in it.

Given set S of N first positive integers, your task is to find the largest subset S' of S such that S' is pairwise coprime.

Input

The first line of input is the number T - the number of tests ($T \leq 1000$). Then T tests follow. Each test will be printed in one line with the number N. ($1 \leq N \leq 10^6$)

Output

For each test, print "$Case\ \#i: p$" where i is the 1-based index of the test and p is the size of the largest subset.

Sample

Sample input	Output for sample input
1 5	Case #1: 4

{1,2,3,5} or {1,3,4,5} are the largest pairwise coprime subset.

Problem E – Binary search tree

A binary tree is a tree data structure where each node has at most two children which are referred to as left child and right child. A binary search tree is a binary tree where each node has a comparable key and must satisfy the following conditions:

- The left sub-tree of a node contains only nodes with keys less than the node's key.
- The right sub-tree of a node contains only nodes with keys more than the node's key.
- The left and the right sub-tree are also binary search tree.

Given a binary tree, your task is to choose a maximal set S of connected nodes in the given tree so that these nodes and the relations (left child, right child, or ancestor) among them form a binary search tree.

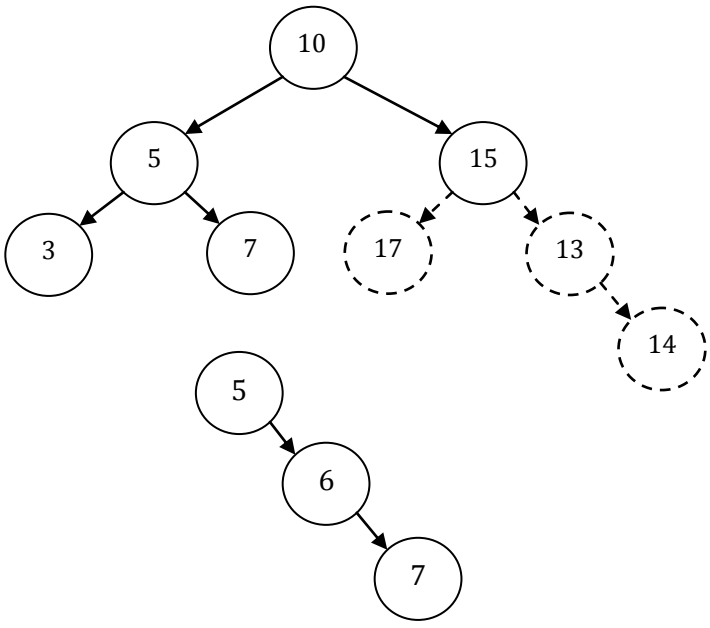
Input

The first line of input is the number T ($T \leq 20$). Then T tests follow. The first line of each test is the number N ($1 \leq N \leq 10^5$) - the number of node of the given binary tree followed by N lines. The i-th line describes node numbered i (nodes are numbered 1 to N). Each node is described by 3 integers: the index of its left child, the index of its right child, the comparable key in that node. The value of a key will not exceed 10^6. If a node doesn't have a left child and/or a right child, they will be represented as 0. The root node is numbered 1.

Output

For each test, print the maximal size of S.

Sample input	Output for sample input
2 8 2 3 10 4 5 5 6 7 15 0 0 3 0 0 7 0 0 17 0 8 13 0 0 14 3 0 2 5 0 3 6 0 0 7	5 3



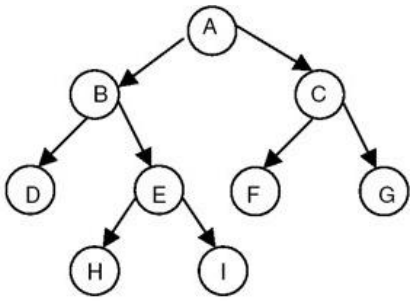
Problem F – Tree again

You are given a directed rooted tree T with N vertices ($N \leq 10^5$). The vertices are numbered from 1 to N, with vertex 1 being the root. You must perform Q queries of two types on the tree ($Q \leq 10^5$):

1. Given 2 vertices u and v, disconnect u from its father and make u the last child of vertex v. For this type of query, it is guaranteed that u is not an ancestor of v.
2. Find the k-th vertex of the pre-order tree traversal.

The pre-order tree traversal can be explained by this pseudo-code and figure.

```
pre_order_travel(node)
{
    print(node)
    for  i = 1 → number of children of  node:
    {
        x = i-th children of  node
        pre_order_travel(x)
    }
}
```



Pre-order traversal: A B D E H I C F G

Input

The first line of input contains the only integer T - number of test cases ($T \leq 10$). Then T tests follow. For each test case:

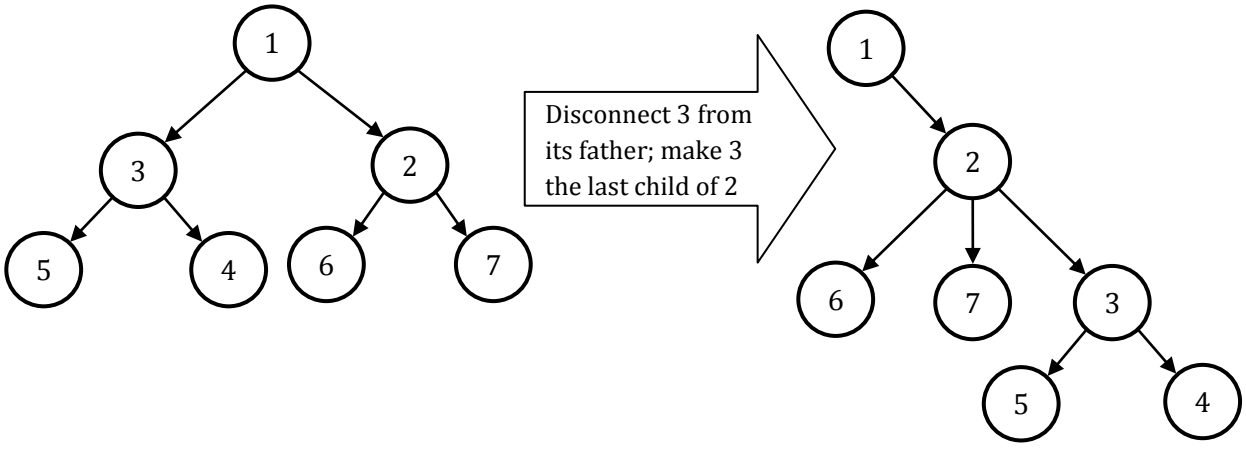
- First line contains the only integer N.
- Next $N-1$ lines, each line contains 2 integers x and y indicating that y is a child of x. The children of x appear in order.
- Next line contains the only integer Q.
- Next Q lines contain descriptions of the queries, in one of the two formats:
  - $1\ u\ v$ ($1 \leq u, v \leq N$ and it is guaranteed that u is not an ancestor of v).
  - $2\ k$ ($1 \leq k \leq N$).

Output

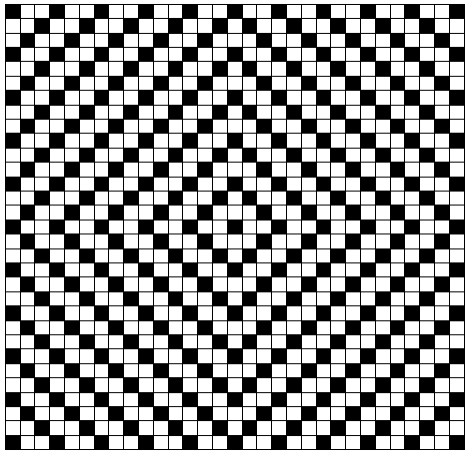
For each query of type 2, print one line containing the only integer - the answer of the query.

Sample

Sample input	Output for sample input
1 7 1 3 1 2 3 5 3 4 2 6 2 7 15 2 1 2 2 2 3 2 4 2 5 2 6 2 7 1 3 2 2 1 2 2 2 3 2 4 2 5 2 6 2 7	1 3 5 4 2 6 7 1 2 6 7 3 5 4



Problem G – Visual illusion



Look closely at this picture! You may think there is curve toward the center because of visual illusion but actually not. It is just a square table with black and white cells in special pattern. This is called visual illusion.

A visual illusion is characterized by visually perceived images that differ from objective reality. The information gathered by the eye is processed in the brain to give a perception that does not tally with a physical measurement of the stimulus source.

This is just a $(2n + 1) \times (2n + 1)$ table with black and white square cells. It starts with an almost white table and one black cell in the center. Then we start coloring cells in steps. In each step, we color some cells so that these cells make a minimal diamond shape and they do not share an edge with each other. The diamond shape in one step should also contain all cells colored in previous steps and do not touch them (sharing a point with them). We continue this process even when part of the diamond shape is out of the table.

Conan found this so interesting; and wants to know if a bigger table would make a stronger illusion. He needs you to write a program to print a whole table of this pattern.

Input

The input starts with T - number of tests ($T \leq 10$). Then T tests follow. Each test is printed in a line consist of a positive integer n. ($1 \leq n \leq 100$)

Output

For each test, print the $(2n + 1) \times (2n + 1)$ table using 'b' or 'w' (denotes a black or white cell).

Sample

Sample input	Output for sample input
2 1 2	www wbw www wbwbw bwwwb wwbww bwwwb wbwbw

Problem H – CCTV

Closed-circuit television (CCTV), also known as video surveillance, is the use of video cameras to transmit a signal to a specific place, on a limited set of monitors. They are widely used for security purposes.

Next week, there will be an exhibition of famous Renaissance artwork. The exhibition room has a rectangular form with the lower left corner at (0,0) and the upper right corner at (X, Y). In 4 corners of the room, 4 CCTVs are used to surveillance every moves of every visitors. Inside this room, N impressive artworks are placed in glass cases. These glass cases are all rectangular shape with edges parallel to the walls; the lower left corner is at (X_{1i}, Y_{1i}) and the upper right corner is at (X_{2i}, Y_{2i}). A CCTV can track any point if the direct line from the point to the CCTV is not blocked by some case.

Given the shape of the exhibition room, the position and shape of all the cases, your task is to calculate the track-able area of 4 CCTV.

Input

The input starts with T ($T \leq 20$) – the number of tests. Then T tests follow. Each test is formatted as:

- The first line consists of X, Y - the upper right corner of the exhibition room. ($0 < X, Y \leq 1000$)
- The second line consist of N - the number of glass cases. ($1 \leq N \leq 20$)
- In the next N lines, the i-th line describes the position of the i-th cabinet. It consists of 4 integers $X_{1i}, Y_{1i}, X_{2i}, Y_{2i}$.
- No cabinet share a point with another cabinet or with any CCTV.

Output

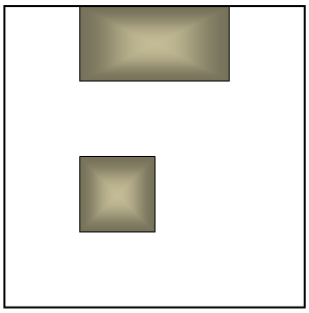
For each test, you should print 2 lines:

- On the first line, print the track-able area of the CCTV at the upper left corner and upper right corner.
- On the second line, print the track-able area of the CCTV at the lower left corner and lower right corner.
- The result should be printed with the precision of less than 1e-4.

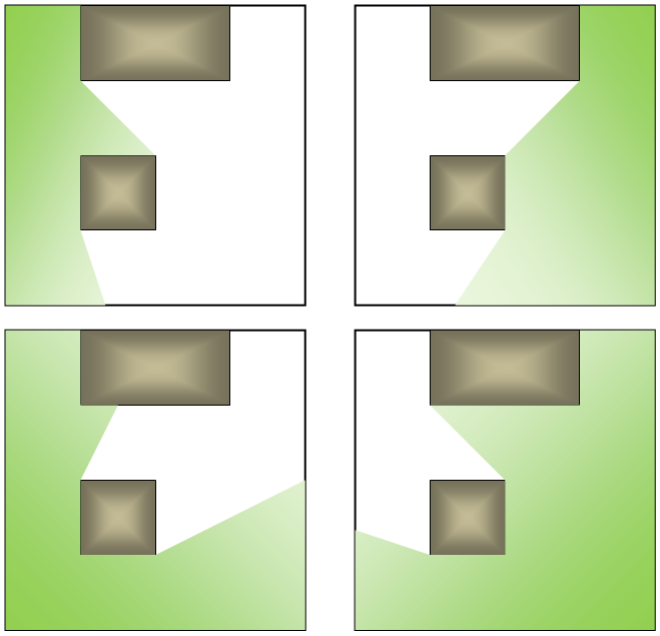
Sample

Sample input	Output for sample input
1 4 4 2 1 1 2 2 1 3 3 4	4.666667 6.833333 8.250000 9.666667

The lay out of the exhibition room in the example:



The track-able area of 4 CCTVs:



Problem I – Fraction

A repeating decimal, also called a recurring decimal, is a number whose decimal representation eventually becomes periodic (i.e., the same sequence of digits repeats indefinitely). The repeating portion of a decimal expansion is conventionally denoted within a pair of brackets so, for example

$$\frac{1}{6} = 0.1666666\ldots = 0.1(6) = 0.1666(6) = 0.166(66)$$

Both 0.1(6) or 0.1666(6) or 0.166(66) are correct representation of 1/6. Given a recurring decimal representation, your task is to find an irreducible fraction that has that representation.

Input

The first line of input contains the number of tests – T ($T \leq 100$). Then T tests follow. Each test is printed in a line as a string whose length does not exceed 15. It is guaranteed to be a meaningful representation of a positive fraction.

Output

For each test, print the result in one line in the format Case #x: a/b

Sample

Sample input	Output for sample input
4 0.125 0.(142857) 0.1(6) .2	Case #1: 1/8 Case #2: 1/7 Case #3: 1/6 Case #4: 1/5

Problem J – Healthy recipes

Over the last few months, Conan has trained very intensively for competing in the ACM/ICPC contest. What he did are just eating, sleeping and coding; as a result, he has put on quite a few kilos. Since the training period are about to finish, he is planning to come back to a healthier life style: going to the gym and eating more properly.

There are N delicious dishes, dish i has cal_i calories. A recipe for a meal is the combination of dishes where each dish appears no more than 1. A perfect, healthy recipe should have the total calories of M calories.

Conan is planning his meals to details and he wonders will there be enough K different perfect recipes for the next K days. 2 recipes are considered different if there is at least a dish appears in one recipe but not the other.

Input

The input starts with the number T - the number of tests. Then T tests follow.

- The first line of each test is 3 integers N, M, K. ($1 \le N \le 100, 1 \le M \le 10000, 1 \le K \le 100$)
- The second line of each test consist of N integers A_i. ($1 \le A_i \le 10000$)

Output

For each test, if there are at least K different perfect recipes, print "ENOUGH"; otherwise you should print the number of perfect recipes.

Sample

Sample input	Output for sample input
2 10 1000 7 100 200 300 400 500 600 700 800 900 1000 10 1000 30 100 200 300 400 500 600 700 800 900 1000	ENOUGH 10

All the dishes in 2 samples are similar. There are 10 different perfect recipes:

$1000 = 100 + 200 + 300 + 400$

$1000 = 200 + 300 + 500$

$1000 = 100 + 200 + 700$

$1000 = 200 + 800$

$1000 = 100 + 300 + 600$

$1000 = 300 + 700$

$1000 = 100 + 400 + 500$

$1000 = 400 + 600$

$1000 = 100 + 900$

$1000 = 1000$