The 37th Annual ACM

International Collegiate Programming Contest

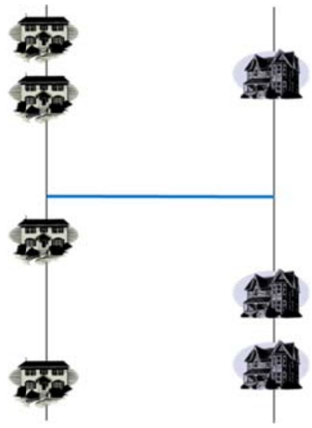
Asia Regional – Daejeon

Nationwide Internet Competition

Problem A
Bridge

A river runs from north to south in a city. Houses are located along each side of the river as illustrated in Figure 1. The city plans to construct a horizontal bridge across the river so that people in both sides can reach quickly each other through the bridge.

Precisely, the left side of the river is the vertical line with the x-coordinate of -1, and the right side of the river is the vertical line with the x-coordinate of $+1$. The vertical strip between two sides represents the river. The bridge is represented as a horizontal segment which connects two points on both sides. The positions of the houses are given on the vertical lines.



e 1.

To determine the optimal location of the bridge, the city want to minimize the sum of the distances between pairs (a, b) of houses where a is in the left side and b is in the right side. The distance between a and b is the sum of the distance from a to the end of the bridge, the length of the bridge, and the distance from the end of the bridge to b. Thus we want to construct the bridge so that the sum of all the distances between houses in both sides is minimized.

More formally, let a_i and b_j be the positions of houses in the left and the right side, respectively, $i = 1, \dots, n$ and $j = 1, \dots, m$ and let h be the position of the bridge. Then the goal is to minimize the following quantity:

$$\sum_{i,j} d(a_i, b_j) = \sum_{\forall i,j} \left(|a_i - h| + 2 + |h - b_j| \right).$$

The positions of houses are given as integers on the vertical lines with the x-coordinates of -1 or $+1$. You should find the position of the bridge, that is, the y-coordinate of the horizontal segment representing the bridge, minimizing the sum of all the distances between the houses in both sides.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with integers n and m, the number of houses of the left and the right side, respectively, where $1 \le n, m \le 1{,}000{,}000$. Each of the following n lines contains an integer a, representing the y-coordinate of a house in the left side. Also each of the following m lines contains an integer b, representing the y-coordinate of a house in the right side. Note that all the y-coordinates of the houses on each side are distinct, and $-10{,}000{,}000 \le a, b \le 10{,}000{,}000$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain a real value, the position of the bridge, that is, the y-coordinate of the horizontal segment minimizing the sum of all the distances between the houses of both sides; your output must contain the first digit after the decimal point, that is, simply ignore the ones from the 2nd digit after the decimal point. If there are multiple solutions, then print the smallest value of them.

The following shows sample input and output for two test cases.

Sample Input	**Output for the Sample Input**
2 3 4 30 -16 5 -5 25 -20 -10 4 7 18 -15 -3 2 8 20 12 -3 18 9 4	-5.0 4.0

Problem B
Canoe Racer

International Canoe Sprint Championship (ICSC) will be held near future. The official boats recognized in the ICSC are the following: C1, C2, and C4, where the number indicates the number of paddlers and "C" stands for canoe. Canoe racing takes place on a straight course divided in lanes, on calm water. The distances recognized in the ICSC for international races are 200m, 500m, and 1000m.

Korea Sports School (KSS) will participate in ICSC for C4 1000m game. KSS has 4 classes of the same number of students, and one student in each class will be selected as the racer of the canoe game. KSS has many C4 type racing boats each of which gives the best performance when the total weight of racers is closest to a specific value. For example, suppose the specific value of a boat is 300 and the weights of students in four classes are as in the following:

Class-1: 60, 52, 80, 40
Class-2: 75, 68, 88, 63
Class-3: 48, 93, 48, 54
Class-4: 56, 73, 49, 75

The four students of weights 60, 75, 93, and 73 in the four classes are the most suitable racers for the boat because their weight sum 301 is closest to 300. In some cases, there can be two different closest weight sums. For example, suppose the specific value of a boat is 200 and there are two closest weight sums 198 and 202. In such case the smaller sum is preferred for the canoe game. So, four students having weight sum 198 should be selected.

Given a specific value of a boat and the weights of students, you are to select four racers satisfying the above condition.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with two integers $\boldsymbol{k}$ and $\boldsymbol{n}$. $\boldsymbol{k}$ is the specific value of a boat and $\boldsymbol{n}$ is the number of students in each class of KSS, where $\mathbf{1 \le \boldsymbol{k} \le 40,000,000}$ and $\mathbf{1 \le \boldsymbol{n} \le 1,000}$. Each of the following four lines contains $\boldsymbol{n}$ integers representing the weights of students in each class of KSS. Each weight is between $\mathbf{1}$ and $\mathbf{10,000,000}$, inclusively.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain an integer representing the weight sum of students who are selected as canoe racers.

The following shows sample input and output for three test cases.

Sample Input	**Output for the Sample Input**
3 300 4 60 52 80 40 75 68 88 63 48 93 48 54 56 73 49 75 8 3 1 2 3 1 2 3 1 2 3 1 2 3 32 2 2 5 9 4 10 20 4 2	301 8 31

Problem C
Castle

A satellite sent us a photo of a ruin in the middle of some desert. The ruin in the photo turns out to an ancient castle of the forgotten kingdom which is 2000 years old. We also know that the floor plane of the castle has a shape of a rectilinear polygon.

A rectilinear polygon is a polygon whose edges are either horizontal or vertical. That is, at each vertex of the polygon, the interior angle formed by its two incident edges is either 90° or 270°, as shown in Figure 1. We say that a rectilinear polygon is *simple* if (1) each vertex is incident to exactly two edges and (2) there are no edges that intersect each other except at their end vertices.

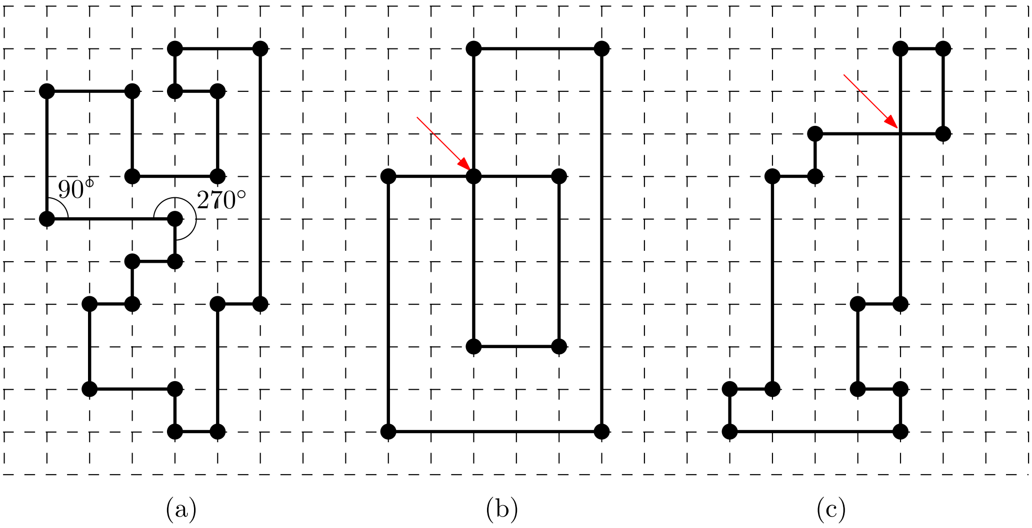
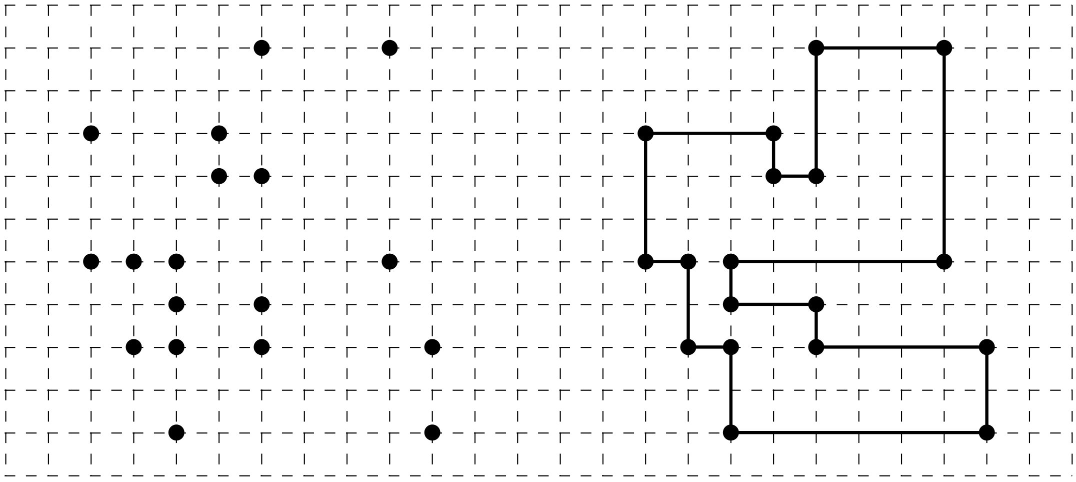


Figure 1. (a) A simple rectilinear polygon. (b)-(c) Non-simple rectilinear polygons because (b) there is a vertex having four incident edges, or (c) there is a pair of edges crossing each other.

The current status of the castle we have figured out from the photo is not good. Only poles of the castle, i.e., the vertices of the simple rectilinear polygon, remained. Thus we need to find out how the poles were connected to recover the original shape of the castle. Figure 2 shows an example.

Formally, you are given a set of n distinct points in the plane with integer coordinates. You need to decide whether or not we can reconstruct a simple rectilinear polygon by connecting all the points with horizontal and vertical segments alternatingly. You should output "YES" if it is possible to reconstruct a simple rectilinear polygon of n vertices from the n input points, "NO" otherwise.



(a) (b)

Figure 2. (a) Input points. (b) A simple rectilinear polygon reconstructed from the input.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with integer $\boldsymbol{n}$, the number of points, where $\mathbf{4} \leq \boldsymbol{n} \leq \mathbf{10,000}$. Each of the following $\boldsymbol{n}$ lines contains two integers, representing $\boldsymbol{x}$-coordinate and $\boldsymbol{y}$-coordinate of the points between $-\mathbf{10^6}$ and $+\mathbf{10^6}$, inclusively. All the points are distinct.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain "`YES`" if the points form a simple rectilinear polygon, "`NO`" otherwise.

The following shows sample input and output for three cases.

Sample Input	Output for the Sample Input
3 18 9 1 8 10 3 1 1 8 4 7 5 10 2 5 3 5 8 5 3 4 1 5 2 3 5 7 5 4 9 3 3 3 5 3 4 8	YES NO NO

6 1 2 2 1 3 2 1 1 3 3 2 3 10 2 5 3 2 2 3 4 4 1 5 3 3 2 2 4 1 2 4 1 1	

Problem D
Fibonacci Numbers

The Fibonacci number f_k is defined by $f_k = f_{k-1} + f_{k-2}$ with seed values $f_0 = 0,\ f_1 = 1$. It is known that every positive integer can be represented as the sum of one or more distinct Fibonacci numbers.
There are many different representations of Fibonacci sums for a positive integer. For example, Fibonacci sums for 100 can be represented by $f_4 + f_6 + f_{11} = 3 + 8 + 89$ or $f_1 + f_3 + f_6 + f_{11} = 1 + 2 + 8 + 89$, or $f_4 + f_6 + f_9 + f_{10} = 3 + 8 + 34 + 55$ etc. In this problem, we want to find the minimum number of Fibonacci numbers for an integer.

For a given positive integer, find the minimum number of distinct Fibonacci numbers whose sum is equal to the given number.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case contains one positive integer n , where $1 \le n \le 1{,}000{,}000{,}000$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the Fibonacci numbers in increasing order which sum is equal to the given number.

The following shows sample input and output for four test cases.

Sample Input	Output for the Sample Input
4 100 200 12345 1003	3 8 89 1 55 144 1 34 377 987 10946 3 13 987

Problem D
피보나치(Fibonacci)

피보나치 수 f_k는 $f_k = f_{k-1} + f_{k-2}$로 정의되며 초기값은 $f_0 = 0$과 $f_1 = 1$ 이다. 양의 정수는 하나 혹은 그 이상의 서로 다른 피보나치 수들의 합으로 나타낼 수 있다는 사실은 잘 알려져 있다.

하나의 양의 정수에 대한 피보나치 수들의 합은 여러 가지 형태가 있다. 예를 들어 정수 100은 $f_4 + f_6 + f_{11} = 3 + 8 + 89$ 또는 $f_1 + f_3 + f_6 + f_{11} = 1 + 2 + 8 + 89$, 또는 $f_4 + f_6 + f_9 + f_{10} = 3 + 8 + 34 + 55$ 등으로 나타낼 수 있다. 이 문제는 하나의 양의 정수를 최소 개수의 서로 다른 피보나치 수들의 합으로 나타내는 것이다.

하나의 양의 정수가 주어질 때, 피보나치 수들의 합이 주어진 정수와 같게 되는 최소 개수의 서로 다른 피보나치 수들을 구하라.

입력(Input)

입력 데이터는 표준입력을 사용한다. 입력은 T 개의 테스트 데이터로 구성된다. 입력의 첫 번째 줄에는 테스트 데이터의 수를 나타내는 정수 T 가 주어진다. 각 테스트 데이터에는 하나의 정수 n이 주어진다. 단, $1 \le n \le 1{,}000{,}000{,}000$.

출력(Output)

출력은 표준출력을 사용한다. 하나의 테스트 데이터에 대한 해를 하나의 줄에 출력한다. 각 테스트 데이터에 대해, 피보나치 수들의 합이 주어진 정수에 대해 같게 되는 최소수의 피보나치 수들을 증가하는 순서로 출력한다.

다음은 네 개의 테스트 데이터에 대한 입력과 출력의 예이다.

입력 예제(Sample Input)	출력 예제(Output for the Sample Input)
4 100 200 12345 1003	3 8 89 1 55 144 1 34 377 987 10946 3 13 987

Problem E
Möbius Strip

The Möbius strip is a surface with only one side and only one boundary edge. Möbius strip can be created as follows: Take a strip of paper and glue the ends together after twisting one end a half turn (see Figure 1).

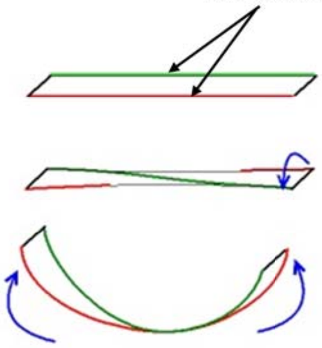


Figure 1. Construction of a Möbius strip

Given a rectangular strip of paper with square grid pattern of size $m \times n$ $(m \leq n)$ on both sides of the strip, we can create a Möbius strip by joining the shorter ends of the strip together. The Möbius strip is said to be of size $m \times 2n$. For example, Figure 2 shows a Möbius strip of size 5×100. Two squares on the Möbius strip are said to be adjacent if they touch each other by a side (except a side on a boundary edge). Consider a small ant lying in a square of the Möbius strip. The ant travels around on the strip by always moving to an adjacent square from the current square. Note that the ant cannot move across the boundary edge of the Möbius strip. The distance between two squares on Möbius strip is defined to be the smallest number of squares an ant moved, except the starting square, when an ant travels from a square to the other. Therefore the distance between two adjacent squares is 1.

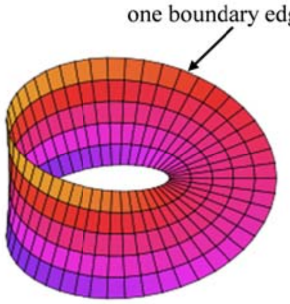


Figure 2. Möbius strip of size 5×100

An average distance of Möbius strip of size $m \times 2n$ is defined to be the average distance between all pairs of squares (including all pairs of the same square) on the strip. For example, the average distance of Möbius strip of size $1 \times 2n$ is $n/2$.

Given a size of Möbius strip, you should output the average distance of the strip.

Input

Your program is to read from standard input. The input consists of $\boldsymbol{T}$ test cases. The number of test cases $\boldsymbol{T}$ is given in the first line of the input. Each test case consists of a single line containing two integers, and $\boldsymbol{m}$ and $(\boldsymbol{1 \leq m \leq n \leq 1{,}000{,}000})$, where the size of input Möbius strip is $\boldsymbol{m \times 2n}$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain a real value, the average distance of the input Möbius strip; the output should have a precision of exactly 1 digit after decimal point (You have to round to the nearest tenth, i.e., the first digit after decimal point).

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 1 100 5 50	50.0 26.6

Problem F
Order

We have a sequence of n integers $S = (s_1,\ s_2,\ \dots,\ s_n)$, where $s_i \neq s_j$ and $1 \leq s_i \leq n$. Using S, we newly define another sequence $R = (r_1,\ r_2,\ \dots,\ r_n)$, where r_i is defined as the number of integers which are smaller than s_i among the integers in $\{s_1, s_2,\ \dots,\ s_{i-2}, s_{i-1}\}$

Let us show one example for $n = 10$. Assume that $S = (6, 4, 3, 5, 1, 2, 7, 8, 9, 10)$, then R should be $R = (0, 0, 0, 2, 0, 1, 6, 7, 8, 9)$. When you are given a sequence R, write a program which reconstructs the original sequence S. In some cases it is impossible to reconstruct S from R. For example, if $n = 5$ and $R = (0, 2, 2, 0, 1)$, then we know that it is impossible to reconstruct S from it.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with an integer n $(1 \leq n \leq 100)$, the number of integers in S. Then a sequence R of n integers is given in the next line.

Output

Your program is to write to standard output. For each sequence R, your program should find the corresponding sequence S from R and print it in each line. If you cannot reconstruct S from R, then your program should print "`IMPOSSIBLE`".

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 10 0 0 0 2 0 1 6 7 6 9 10 0 0 0 0 0 0 0 0 0 0 12 0 3 4 5 0 1 2 3 4 5 6 7	6 4 3 5 1 2 8 9 7 10 10 9 8 7 6 5 4 3 2 1 IMPOSSIBLE

Problem F
순서(Order)

n 개의 정수로 된 순서 $S = (s_1, s_2, \ldots, s_n)$가 있다. 여기서 $s_i \neq s_j$이고 $1 \leq s_i \leq n$이다. S로부터 새로운 순서 $R = (r_1, r_2, \ldots, r_n)$을 얻을 수 있는데, 여기서 r_i는 S의 부분순서 $\{s_1, s_2, \ldots, s_{i-2}, s_{i-1}\}$ 중에서 s_i보다 작은 수의 개수이다.

$n = 10$인 경우의 예를 보자. $S = (6, 4, 3, 5, 1, 2, 7, 8, 9, 10)$이라면 $R = (0, 0, 0, 2, 0, 1, 6, 7, 8, 9)$이 된다. 어떤 순서 R이 주어질 때, 여러분은 R을 이용하여 S를 찾는 프로그램을 작성하여야 한다. 경우에 따라서는 R로부터 S를 찾는 것이 불가능할 수 있다. 예를 들어, 만약 $n = 5$이고, $R = (0, 2, 2, 0, 1)$이라면 이런 R에 대응하는 S는 존재하지 않는다.

입력(Input)

입력 데이터는 표준입력을 사용한다. 입력은 T개의 테스트 데이터로 구성된다. 입력의 첫 번째 줄에는 입력 데이터의 수를 나타내는 정수 T가 주어진다. 각 테스트 데이터의 첫째 줄에는 S에 있는 정수의 개수 n $(1 \leq n \leq 100)$이 주어진다. 그 다음 줄에는 n개의 정수로 된 R이 주어진다.

출력(Output)

출력은 표준출력을 사용한다. 주어진 각 순서 R에 대해, 대응하는 순서 S를 찾아 한 줄에 출력하여야 한다. 만약 R로부터 S를 찾는 것이 불가능할 경우에 "`IMPOSSIBLE`"을 출력한다.

다음은 세 개의 테스트 데이터에 대한 입력과 출력의 예이다.

입력 예제(Sample Input)

```
3
10
0 0 0 2 0 1 6 7 6 9
10
0 0 0 0 0 0 0 0 0 0
12
0 3 4 5 0 1 2 3 4 5 6 7
```

출력 예제(Output for the Sample Input)

```
6 4 3 5 1 2 8 9 7 10
10 9 8 7 6 5 4 3 2 1
IMPOSSIBLE
```

The 37th Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Nationwide Internet Competition

Problem G
Parenthesis

Parenthesis String (PS) consists of two parenthesis symbols '(' and ')' only. In parenthesis strings, some strings are called a *valid* PS (shortly, VPS). Let us give the formal definition of VPS. A single "( )" is a member of VPS, called the base VPS. Let x and y be a member of VPS. Then "(x)", a VPS which encloses a VPS x with a single pair of parenthesis, is also a member of VPS. And xy, the concatenation of two VPS x and y, is a member of VPS. For example, "(( )) ( )" and ((( )))" are all VPS, but "(( ) (", "(( )) ( )))" and "(( )" are not VPS. You are given a set of PS. You should decide if the input string is VPS or not.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Then PS's are given in the following T lines one by one. The length of each PS is between 2 and 50, inclusively.

Output

Your program is to write to standard output. Print the result in each line. If the input string is a VPS, then print "YES". Otherwise print "NO".

The following shows sample input and output for 6 test cases.

Sample Input	Output for the Sample Input
6	NO
(())())	NO
(((()())()	YES
(()())((()))	NO
((()()(()))(((())))()	YES
()()()()(()()())()	NO
(()((())()(	

Problem G

괄호(Parenthesis)

괄호 문자열(Parenthesis String, PS)은 두 개의 괄호 기호인 '(' 와 ')' 만으로 구성되어 있는 문자열이다. 그 중에서 괄호의 모양이 바르게 구성된 문자열을 올바른 괄호 문자열(Valid PS, VPS)이라고 부른다. 한 쌍의 괄호 기호로 된 "( )" 문자열은 기본 VPS 이라고 부른다. 만일 x 가 VPS 라면 이것을 하나의 괄호에 넣은 새로운 문자열 "(x)"도 VPS 가 된다. 그리고 두 VPS x 와 y를 접합(concatenation)시킨 새로운 문자열 xy도 VPS 가 된다. 예를 들어 "(( )) ( )"와 "((( )))" 는 VPS 이지만 "(( ) (", "(( )) ( )))" , 그리고 "(( )" 는 모두 VPS 가 아닌 문자열이다.

여러분은 입력으로 주어진 괄호 문자열이 VPS 인지 아닌지를 판단해서 그 결과를 YES 와 NO 로 나타내야 한다.

입력(Input)

입력 데이터는 표준 입력을 사용한다. 입력은 T개의 테스트 데이터로 주어진다. 입력의 첫 번째 줄에는 입력 데이터의 수를 나타내는 정수 T가 주어진다. 각 테스트 데이터의 첫째 줄에는 괄호 문자열이 한 줄에 주어진다. 하나의 괄호 문자열의 길이는 2 이상 50 이하이다.

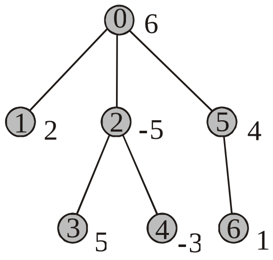
출력(Output)

출력은 표준 출력을 사용한다. 만일 입력 괄호 문자열이 올바른 괄호 문자열(VPS)이면 "YES", 아니면 "NO"를 한 줄에 하나씩 차례대로 출력해야 한다.

다음은 6 개의 데이터를 가지는 입력과 출력의 예를 보여주고 있다.

Sample Input	Output for the Sample Input
6 (( )) ( )) (((( ) ( )) ( ) (( ) ( )) ((( ))) ((( ) ( ) (( ))) (((( )))) ( ) ( ) ( ) ( ) ( ) (( ) ( ) ( )) ( ) (( ) ((( )) ( ) (	NO NO YES NO YES NO

Problem H
Parents



Consider a rooted tree, each of whose nodes is associated with an integer representing the node's value. Naturally, *the value of a subset of nodes* of such a tree is regarded as the sum of the values of all the nodes contained in the subset. Given a positive integer K, let's define *the K-anti-parental subset* to be a subset of nodes satisfying the following constraints: (1) the number of nodes in the subset is between 1 and K, inclusive, and (2) for any pair of nodes in the subset, one is not a parent of the other, and vice versa. Now, your task is to find the maximum of the values of the K-anti-parental subsets of an arbitrary node-valued tree.

Input

Your program is to read from standard input. The input consists of T test cases, where the positive integer T is given in the first line of the input, followed by the description of each test case. The first line of a test case contains two positive integers N and K, respectively indicating the number of nodes of the tree and the parameter K as explained above, in which we assume $N \leq 100{,}000$ and $1 \leq K \leq 100$. The tree's nodes are indexed 0 to $N-1$, where the index 0 is always assigned to the root node. The following line contains N integers, separated by spaces, coming from the inclusive interval $[-1{,}000, 1{,}000]$, which represent the values of the nodes enumerated in the increasing order of indices from 0 to $N-1$. The next following line contains $N-1$ integers, separated by spaces, each of which indicates the index of a corresponding node's parent. Make sure that these numbers are for the $N-1$ nodes except the root, listed in the increasing order of indices from 1 (not 0) to $N-1$. Therefore, the first integer points to the parent of the node indexed 1. Note that the parent of the root node needs not be specified.

Output

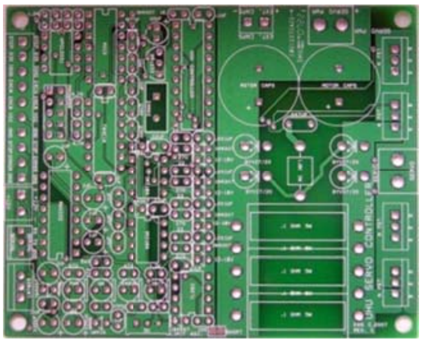
Your program is to write to standard output. Print exactly one line per each test case. The line should contain the maximum possible value of the K-anti-parental subsets of the input tree.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 7 3 6 2 -5 5 -3 4 1 0 0 2 2 0 5 3 2 4 -3 5 0 0 7 6 -1 -1 -1 -1 -1 -1 -1 0 1 2 3 4 5	12 5 -1

Problem I
PCB

A Printed Circuit Board (shortly, PCB) is used to electrically connect electrical components using conductive pathways (or wires) etched from copper sheets laminated onto a non-conductive substrate. Typical PCBs can be easily seen inside any personal computer, including a mainboard, a graphic card, and RAMs. For many reasons, a PCB should connect electrical components on it in a compact and efficient way, so how to wire each component to others is a central issue in designing PCBs.



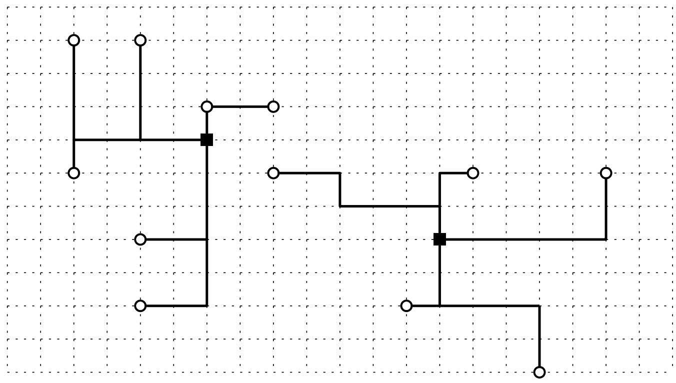
International Circuit Production Corporation (shortly, ICPC), which is a company that manufactures PCBs on orders, recently received an order of designing a PCB, model code iCPC-2012, which will be installed in a next generation smart mobile device. According to the specification of iCPC-2012, it consists of N components $C_1, \ldots, C_N$ and two special components, called *clocks*. A clock can send signals periodically to at most K ($\geq N/2$) components that are connected to it; that is, a clock can serve at most K components. Because all the N components in iCPC-2012 are required to be synchronized during their operations, each component C_i has to be connected to one of the two clocks. The two clocks are supposed to be perfectly synchronized.

ICPC has so far decided the shape of the board of iCPC-2012 and the locations of each of the N components on it, but not where to put two clocks. In a viewpoint of synchronization among components, ICPC wants to minimize the maximum length of pathways from each component C_i to its connected clock. You are to write a program that computes the minimum possible length of the longest pathway between components and their connected clock when the two clocks are optimally located.

More specifically, iCPC-2012 has the following properties.

(i) Every pathway between a component and a clock consists of horizontal or vertical segments.
(ii) Every pathway runs under the components. So, you can freely design pathways, making them as short as possible.
(iii) The location $p_i = (x_i, y_i)$ of each component C_i is given as a pair of *even* integers, representing the coordinates on the board.
(iv) Each clock can be located at any place (x, y) on the board with $-10^6 \leq x, y \leq 10^6$. Clocks even can be located at the same location p_i of component C_i.

Remark by (i) and (ii) that the length of the pathway from C_i to its connected clock is determined by the locations of the component and the clock; not by how to connect them.



An example is illustrated above, where $N = 12$ and $K = 7$. The locations p_i of the 12 components are depicted as small circles and an optimal location of two clocks are marked as black squares. As shown above, the pathways, depicted by segments, fulfill (i) and (ii), and the length of longest pathways is 7, which will be the correct answer of your program. Note that this optimal location of clocks is not unique.

Input

Your program is to read from standard input. The input consists of $\boldsymbol{T}$ test cases. The number $\boldsymbol{T}$ of test cases is given in the first line of the input. From the second line, each test case is given in order, consisting of the following: a test case contains two integers $\boldsymbol{N}$ ($\mathbf{2} \le \boldsymbol{N} \le \mathbf{100,000}$) and $\boldsymbol{K}$ ($\boldsymbol{N}/\mathbf{2} \le \boldsymbol{K} \le \mathbf{100,000}$) in its first line, and is followed by $\boldsymbol{N}$ lines each of which consists of two even integers inclusively between $-\mathbf{1,000,000}$ and $\mathbf{1,000,000}$, representing the $\boldsymbol{x}$- and $\boldsymbol{y}$-coordinates of $\boldsymbol{p_i} = (\boldsymbol{x_i}, \boldsymbol{y_i})$, the location of $\boldsymbol{C_i}$ on the board. Two consecutive integers in one line are separated by a single space and there is no empty line between two consecutive test cases.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain a single integer obtained by rounding off the minimum possible length of the longest pathway. For examples, if the result you obtain is 5.52, then you should print out “6”; if the result is 5.49, then you print out “5”.

The following shows sample input and output for two test cases.

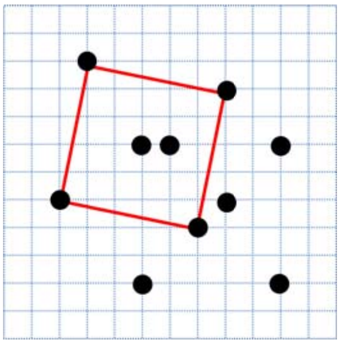
Sample Input	Output for the Sample Input
2 12 7 2 6 2 10 4 2 4 4 4 10 6 8 8 6 8 8 12 2 14 6 16 0 18 6 5 3 0 0 0 2 2 0 2 2 6 6	7 4

The 37th Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Nationwide Internet Competition

Problem J
Square

There are n points in the plane.

Write a program which finds a largest square such that each of its vertices is one of the given points and its area is greater than 0.



Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with an integer n, the number of points, where $4 \leq n \leq 3{,}000$. Each of the following n lines contains two integers, representing x-coordinate and y-coordinate of the points between -10^4 and $+10^4$, inclusively.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain an integer, the area of a largest square. If there is no square, print out 0.

The following shows sample input and output for one test case.

Sample Input

```
1
10
5 2
10 2
7 4
2 5
8 5
5 7
6 7
10 7
8 9
3 10
```

Output for the Sample Input

```
26
```