The 2015 Asia ACM-ICPC Hanoi Regional Contest

Problems Overview

Problem A: Obfuscated Emails

Problem B: Parallelogram

Problem C: Egyptian Encryption

Problem D: Work Effectiveness

Problem E: Pepsi Distribution

Problem F: Genome

Problem G: Cloud Computing

Problem H: Salary Adjustment

Problem I: Traffic Network

Problem J: Playground

Problem K: Tree

Problem L: Stick Throwing

Note: The input and output for all the problems are standard input and output.

Problem A: Obfuscated Emails

Celebrating its 10 years anniversary, acmicpc.edu.vn website has just announced a lucky draw for their users. ***N*** users have been selected randomly to receive a special prize. They will announce ***N*** winners' emails. However, stating the full emails of the winners in public is not a good idea because they might receive a lot of spam emails. Thus, the website will only publish the obfuscated emails.

An email consists of 3 parts: a username with a minimal of 3 characters, a special character @ and the mail server domain. For example, in an email "**webmaster@acmicpc.edu.vn**", **"webmaster"** is the username, "**acmicpc.edu.vn**" is the domain.

To obfuscate this email, we will keep the first and last character of the username, replace other characters in the username with character * (the asterisk, ASCII 42); the remaining part of the email stays intact. Thus "**webmaster@acmicpc.edu.vn**" will become "**w*******r@acmicpc.edu.vn**".

Your task is to obfuscate the emails of ***N*** winners.

Input

The input starts with a number ***N*** (***N*** ≤ 20) - the number of winners on a single line, followed by ***N*** winners' emails where each email is on a separate line. These emails are guaranteed to be valid (only consists of lowercase letters, digits, the dot or full stop and the character @).

Output

For each email in the input, print the obfuscated version in one line.

Sample Input	Sample Output
2 webmaster@acmicpc.edu.vn hanoiregional2015@hubt.edu.vn	w*******r@acmicpc.edu.vn h***************5@hubt.edu.vn

Problem B: Parallelogram

In Euclidean plane geometry, a parallelogram is a (non self-intersecting) quadrilateral with two pairs of parallel sides. The length of opposite or facing sides of a parallelogram are equal and the opposite angles of a parallelogram are equal.

Given N distinct points, your task is to find the number of parallelograms that can be formed by any 4 points of these N points.

Input

The input consists of several data sets. The first line of the input contains the number of data sets, which is a positive number and is not bigger than 20. The following lines describe the data sets.

Each data set is described by the following lines:

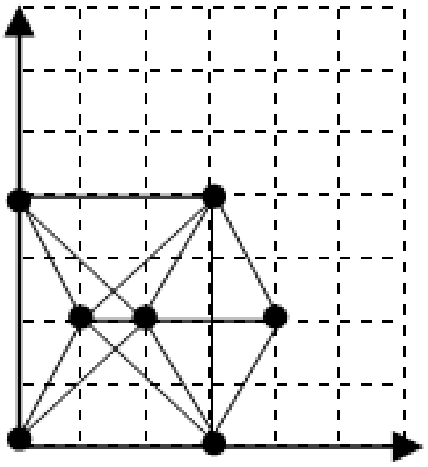
- The first line contains an integer N ($N \le 1000$).
- The i^{th} line of the following N lines contains two integers x_i, y_i that are the coordinates of the i^{th} point ($|x_i|$, $|y_i| \le 10^9$).

Two consecutive numbers on the same line are separated by a space.

Output

For each data set, write in one line the number of parallelograms that can be formed.

Sample Input	Sample Output	Illustration

1
7
0 0
3 0
4 2
1 2
2 2
0 4
3 4
	6	

Problem C: Egyptian Encryption

Archaeologists have found an old Egyptian document mentioning the location of a treasure. However, this document was encrypted. Fortunately, based on ancient documents, the archaeologists know that the document was encrypted by simply permuting the order of the words. The encryption is conducted by an encoder, which is a permutation of word indexes from 1 to document length (i.e. the number of words in the document). These indexes indicate the positions of corresponding words after one encoding. For example, if the original word order is **a b c d**, with an encoder 3 1 2 4, the encoded document is **c a b d**.

Interestingly, ancient Egyptian mathematicians had found out that we could actually receive the original document after several times of successively encoding the document using a given encoder. They called the number of successive encoding needed to get back the original document the encryption length of the encoder. For example, suppose a document has a length of 4 and the word positions are **a b c d**. Given an encoder 2 3 1 4, we have:

The original document	**a b c d**
The document after one encoding	**c a b d**
The document after two successive encoding	**b c a d**
The document after three successive encoding	**a b c d**

Based on the mathematicians' advice, Pharaoh - the king of Egypt - decided to use the encoder having the maximum encryption length to encrypt all his documents. Your task is to help them find the maximum encryption length of the encoder used for a document having length ***L***.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets. Each dataset contains one line with a positive integer ***L*** representing the number of words of the encrypted document ($2 \le L \le 10^5$).

Output

For each data set, write in one line the remainder of the maximum encryption length divided by 2147483647 which is (2^{31} -1)

Sample Input	Sample Output
2 4 5	4 6

Problem D: Work Effectiveness

The software company organizes an “Advanced Algorithms” class to improve the programming skill of programmers in the company. There are $2N$ programmers attending the class. In the practice hours, the instructor wants to divide the students into N pairs and to assign a programming task to each pair. The instructor has identified a measure for the work effectiveness of a pair of programmers i and j (which is the same as the work effectiveness for the pair of programmers j and i), which is represented by an integer a_{ij}. The work effectiveness of the whole class is defined as the smallest work effectiveness among all work effectiveness of N student pairs.

Given a_{ij} (i, j = 1, 2,…, $2N$), your task is to find the maximum work effectiveness when distributing $2N$ programmers into N pairs.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets.

Each dataset contains the following information:

- The first line contains an integer N ($2 \le N \le 200$).
- The i^{th} line of the $2N$-1 following lines contains $2N-i$ integers $\mathbf{a}_{i,i+1}$, $\mathbf{a}_{i,i+2}$, …, $\mathbf{a}_{i,2N}$ (i = 1, 2,…, $2N$-1). It is also given that $\mathbf{a}_{ij} = \mathbf{a}_{ji}$ and $0 \le \mathbf{a}_{ij} \le 10^{15}$ (i, j = 1, 2,…, $2N$-1).

Two consecutive numbers on the same line are separated by a space.

Output

For each data set, write in one line an integer that is the maximum work effectiveness of the class.

Sample Input	Sample Output

1
2
5 7 8
7 5
4
	7

Problem E: Pepsi Distribution

A sales-truck has a mission of distributing Pepsi packages on a specified route, which is a straight line. The sales-truck departs from point **A**, which has coordinate 0, and heading toward positive coordinates. The sales-truck has a limited capacity in the sense that it can carry at most ***C*** Pepsi packages at any moment. On the route, there are ***N*** clients and ***M*** warehouses. Each client locates at a positive integer coordinate, and requires a specific number of Pepsi packages. Each warehouse locates at a positive integer coordinate. No client or warehouse locates at the same coordinate. Each warehouse has an unlimited number of Pepsi packages that the sales-truck can load for distributing to clients. The sales-truck travels and distributes Pepsi packages to the clients with the following rules:

- From the departure ***A*** or a warehouse, the sales-truck fully loads its container with Pepsi packages (after loading, it has ***C*** packages).
- At any client *i*, if the number of packages on the sales-truck is greater than or equal to the number of packages D_i required by the client, then the sales-truck distributes D_i packages to the client. Otherwise, the sales-truck ignores the client and continues traveling.
- The sales-truck never travels backward (from a point to another point with smaller coordinate).

Your task is to compute how many Pepsi packages the sales-truck has distributed on the specified route.

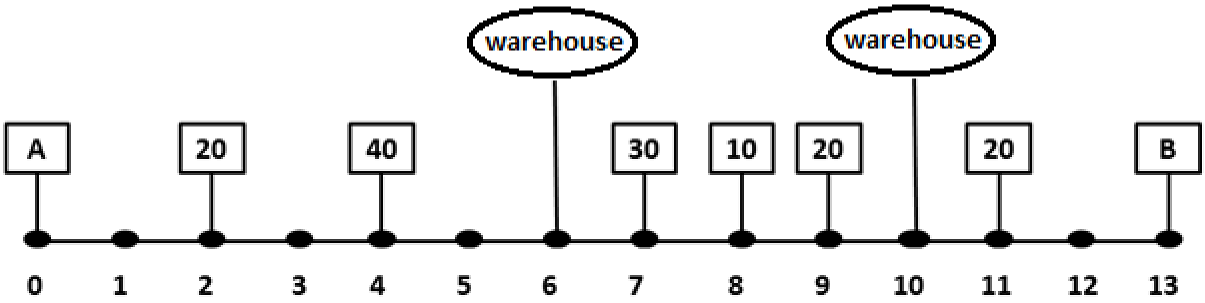


Figure 1. Illustration of Pepsi distribution by a sales-truck

Example: Let's consider the example on Figure 1. The capacity of the sales-truck is **C** = 50 (packages). There are two warehouses at coordinate 6 and 10. There are 6 clients with the following information:

- Client 1: at coordinate 4 and requires 40 packages
- Client 2: at coordinate 8 and requires 10 packages
- Client 3: at coordinate 2 and requires 20 packages
- Client 4: at coordinate 11 and requires 20 packages
- Client 5: at coordinate 9 and requires 20 packages
- Client 6: at coordinate 7 and requires 30 packages

The distribution process is as follows. The sales-truck distributes 20 packages to client 2 and moves to client 1. At this point, the number of remaining packages on the sales-truck's container is 30, which is less than the number of packages (40) required by client 1. Hence, the sales-truck ignores the client 1 and continues to go to the next warehouse at point 6. At this point, the sales-truck fully loads its container (after loading, it has 50 packages on its container). It then distributes 30 packages to client 6 and then distributes 10 packages to the client 2. The number of remaining packages on the sales-truck's container is now 10, which is less than what is required by the client 5. Hence the sales-truck ignores client 5 and continues to go to the warehouse at point 10 for loading. Finally, the sales-truck distributes 20 packages to client 4. The total number of packages that the sales-truck has distributed is 20 + 30 + 10 + 20 = 80.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 10. The following lines describe the datasets. Each dataset contains the following information:

- The first line contains 3 positive integers $\boldsymbol{C}$, $\boldsymbol{M}$, and $\boldsymbol{N}$ in which $\boldsymbol{C}$ is the capacity of the sales-truck, $\boldsymbol{M}$ is the number of warehouses, and $\boldsymbol{N}$ is the number of clients. ($0 < \mathbf{C} \le 10^9$, $0 < \mathbf{N}+\mathbf{M} \le 10^6$)
- The second line contains $\boldsymbol{M}$ positive integer representing the coordinates of $\boldsymbol{M}$ warehouses.
- i^{th} line in the next $\boldsymbol{N}$ lines: contains $\boldsymbol{X}_i$ and $\boldsymbol{D}_i$ which are the coordinate and the number of packages required by client i respectively, $i = 1,\ldots, \boldsymbol{N}$. ($0 < \boldsymbol{X}_i, \boldsymbol{D}_i \le 10^9$)

Two consecutive numbers on the same line are separated by a space.

Output

For each data set, write in one line the number of pepsi packages distributed.

Sample Input	**Sample Output**

2
50 2 6
6 10
4 40
8 10
2 20
11 20
9 20
7 30
2 2 3
2 4
1 2
3 2
5 2
	80
	6

Problem F: Genome

Genome is a sequence of nucleotides. There are four nucleotide types abbreviated by four letters A, C, G, and T. Let **g** = (g_1, g_2,…, g_L) be a genome with L nucleotides (L ≤ 10^5). A *short read* is a sequence of 20 nucleotides. Let ***S*** = (s_1, s_2,…, s_{20}) be a short read. We say that short read ***S*** matches genome ***g*** at position *i* (*i* + 19 ≤ L) if there is at most one position *j* (1 ≤ *j* ≤ 20) where $s_j \neq g_{i+j-1}$.

The *unique level* of short read ***S*** is calculated as the number of positions where short read ***S*** matches genome ***g***. Given a genome ***g*** and a list of *n* short reads $\boldsymbol{S}_1$, $\boldsymbol{S}_2$,…, $\boldsymbol{S}_n$, your task is to calculate the unique level for each of these short reads.

Input

The first line of the input contains an integer *n* (*n* ≤ 10^4).

The i^{th} line of the next *n* lines contains short read $\boldsymbol{S}_i$.

The last line contains genome ***g***.

Output

For each data set, the output should contain *n* lines where the i^{th} line contains the unique level of short read $\boldsymbol{S}_i$.

Sample Input	Sample Output
3 GTTTTTTTTTTTTTTTTCCC ATTTTTTTTTTTTTTTTTTT TTTTTTTTTTTTTTTTTCCC ATTTTTTTTTTTTTTTTTTTTTTTCCC	1 5 2

Problem G: Cloud Computing

Cloud Computing has become a new trend for information processing. Elasticity and scalability are important features of a cloud computing system. When you deploy your application on a cloud computing system, you can easily adjust the number of virtual computers on the cloud for your application.

There are N applications deployed and running on ***A***daptive ***C***loud ***M***odel (ACM) system. The maximum total number of virtual computers that can be executed at a moment on ACM system is 1000.

Today, the i^{th} application is running on $\boldsymbol{A}_i$ virtual computers. However, the number of virtual computers needed for the i^{th} application tomorrow will be $\boldsymbol{B}_i$.

There are three types of operations for ACM system to re-allocate virtual computers for applications:

- **Start** one more virtual computer for the i^{th} application with the cost of $\boldsymbol{X}$.
- **Stop** one virtual computer that is being used by i^{th} application with the cost of $\boldsymbol{Y}$.
- **Transfer** one virtual computer from the i^{th} application to the j^{th} application with the cost of $\boldsymbol{Z} \times |i - j|$.

Your task is to help ACM system to re-allocate virtual computers to meet the actual needs for tomorrow of all applications with minimum cost.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets.

Each dataset contains the following information:

- The first line contains the number of applications $\boldsymbol{N}$ ($1 \le \boldsymbol{N} \le 1000$) and the costs for three types of operations: $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ ($0 \le \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \le 1000$).
- The second line contains N non-negative integers $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_N$. The sum of all $\boldsymbol{A}_i$ is not greater than 1000.
- The third line contains N non-negative integers $\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_N$. The sum of all $\boldsymbol{B}_i$ is not greater than 1000.

Two consecutive numbers on the same line are separated by a space.

Output

For each data set, write in one line an integer that is the minimum cost to re-allocate virtual computers to meet tomorrow needs of all applications.

Sample Input	Sample Output

1
5 200 300 1
5 4 3 2 1
4 3 2 4 1
	303

Explanation:

- Shut down one virtual computer of the first application with the cost of 300.
- Transfer one virtual computer from the second application to the fourth application with the cost of 2.
- Transfer one virtual computer from third application to the fourth application with the cost of 1.

Problem H: Salary Adjustment

A multi-national company exercises the following salary structure. There are $\mathbf{n}+2$ salary levels which have the values s_0, s_1, s_2,…, s_{n+1}, where s_0 is smaller than others, s_{n+1} is greater than others, and s_i is not necessarily greater than s_j for $i > j$ (i = 1..**n**, j = 1..**n**). Currently, there are n employees where employee i has a salary level i, i = 1...n.

At the end of every month, the Reward committee and Disciplinary committee will independently analyze the performance of every employee. Person i (i = 1..n) will get an increase of p_i steps from the Reward committee and a decrease of q_i steps from the disciplinary committee ($0 \le p_i, q_i \le n$).

A step of increasing salary level is conducted as follows: an employee currently at level i will be moved to the closest level $k > i$ such that $s_k > s_i$. If the salary level reaches the maximum salary then further salary increase will have no effect.

A step of decreasing salary level is conducted as follows: an employee currently at level i will be moved to the closest level $r < i$ such that $s_r < s_i$. If the salary level reaches the minimum salary then further salary decrease will have no effect.

Every employee can make a choice of choosing whether to apply the decision of the Reward committee before the decision of the Disciplinary committee or the other way around. Everyone wants to have the maximum possible income after applying the decision from the two committees.

Your task is to figure out the salary level of all employees at the end of the month assuming everyone makes optimal decision in maximizing his/her income.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets.

Each dataset contains 04 lines:

- The first line contains an integer n ($1 \le n \le 2\times10^5$).
- The second line contains n integers $s_1, s_2,.., s_n$ ($0 < s_i \le 10^9$, i = 1..n),
- The third line contains n integers $p_1, p_2,.., p_n$ ($0 < p_i \le 10^9$, i = 1..n),
- The fourth line contains n integers $q_1, q_2,.., q_n$ ($0 < q_i \le 10^9$, i = 1..n),

Two consecutive numbers on the same line are separated by a space.

Output

For each data set, write in one line n space-separated integers where the i^{th} integer represents the new salary level of employee i.

Sample Input	Sample Output

1
10
6 2 4 8 5 1 4 9 8 7
3 4 2 1 4 2 3 1 2 2
2 1 4 2 3 3 1 3 1 1
	8 11 4 3 11 4 11 1 11 11

Problem I: Traffic Network

While ACM/ICPC is being held in Hanoi, the organizer has decided to have a satellite competition for designing the future traffic network for Hanoi. Each team submits a simple undirected graph modeling the future traffic network where nodes represent intersections and an edge connecting two nodes models a road between two corresponding intersections.

After thorough consideration, the judging committee has decided to award prizes to teams from HUBT University among $\boldsymbol{T}$ participating teams. An outstanding common feature from the prize holders that other teams do not have is "*For every cycle of 4 or more nodes, there is an edge that is not part of the cycle but connect 2 nodes of the cycle*".

After the competition, the organizers passed all submitted networks to Hanoi but unfortunately all information about the participating teams, including who the winners are, was lost.

Your task is to help the organizers figure out which ones are the winners based on the submitted traffic networks.

Input

The input file consists of $\boldsymbol{T}$ datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 50. The following lines describe the datasets.

Each data set contains the following information:

- The first line contains two integers $\boldsymbol{n}$ and $\boldsymbol{m}$ ($\boldsymbol{n} \le 10^5$, $\boldsymbol{m} \le 2\times10^5$) representing the number of nodes and the number of edges of the network;
- In the next $\boldsymbol{m}$ lines, each line contains two positive integers not greater than $\boldsymbol{n}$ representing two nodes of an edge in the network;
- It is guaranteed that there are no more than one edge between 2 nodes **u** and **v** and there is no edge connecting a node to itself.

Two consecutive numbers on the same line are separated by a space.

Output

For each data set, write in one line number 1 if the graph is a design of a winner or 0 otherwise.

Sample Input	Sample Output

2
6 8
1 2
2 3
3 4
4 5
5 6
6 2
3 6
	1
	0

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

Problem J: Playground

Young kids are very active and always love to play with others. There are N children in the playground and they are playing a game. In each match, one child will be the referee (and cannot play in that match), only $N - 1$ other children will play. The i^{th} child wants to play at least A_i matches. Determine the minimum number of matches to satisfy the desire to play of all young kids.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 25. The following lines describe the datasets.

Each dataset contains the following information:

- The first line contains an integer N ($3 \leq N \leq 10^5$).
- The second line contains N positive integers $A_1, A_2, \ldots, A_N$ where $1 \leq A_i \leq 10^9$.

Two consecutive numbers on the same line are separated by a space.

Output

For each data set, write in one line the corresponding minimum number of matches.

Sample Input	Sample Output

1
3
2 2 3
	4

Problem K: Tree

You are given a tree with n nodes indexed from 1 to n. The distance between two nodes is defined to be the number of edges on the path between them. Let's define the following two actions:

- Change(x, y): ($1 \leq x, y \leq n$; $x \neq y$)
    - If x is in the subtree rooted at y, swap x and y.
    - If x is not in the subtree rooted at y, move the subtree rooted at y to be a branch of x. y is now a child of x.
- Query(x, y, z): ($1 \leq x, y, z \leq n$; $x \neq y$; $y \neq z$; $z \neq x$): find node t such that the sum of distances between x, y, z and t is minimum.

Given m actions, your task is to implement all the given actions in order and give the answer to all Query actions.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 50. The following lines describe the datasets.

Each dataset contains the following information:

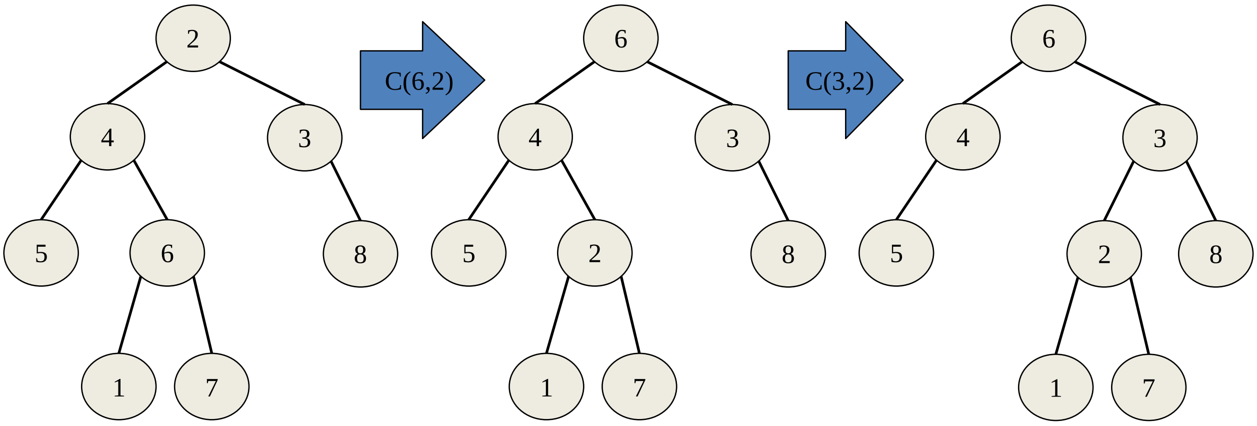
- The first line contains two positive integers n, m ($3 \leq n \leq 10^5$; $1 \leq m \leq 10^5$).
- The second line contains n integers, where the i^{th} number is the index of the parent node of node i; If node i is the root, the i^{th} number will be 0.
- In the next m line, each line contain one action in the following format:
    - C x y: represents action Change(x, y)
    - Q x y z: represents action Query(x, y, z)

Two consecutive numbers on the same line are separated by a space.

Output

For each data set, write in one line the result of each Query action.

Sample Input	Sample Output
1 8 4 6 0 2 2 4 4 6 3 C 6 2 C 3 2 Q 1 8 7 Q 2 5 8	2 3


2
4
3
5
6
8
1
7
C(6,2)
6
4
3
5
2
8
1
7
C(3,2)
6
4
3
5
2
8
1
7

Problem L: Stick Throwing

Little Long is playing a game called "stick throwing". This game is played on an unbounded two-dimensional plane. In the plane, there are unlimited evenly distributed horizontal lines where the distance between 2 adjacent lines is $\boldsymbol{R}$. Little Long has only one chance to throw a stick of length $\boldsymbol{D}$ on the plane. The number of points he score equals to the number of horizontal lines the stick intersects after landing.

Little Long is a beginner in this game and he has no skill of throwing, thus the landing position of his stick is totally random. Little Long wants to know what is the probability $\boldsymbol{P}$ of scoring a positive point (i.e. the stick intersects at least one horizontal line) and what is the expected point $\boldsymbol{E}$ he will get (i.e. the expected number of horizontal line the stick intersects)?

Your task is to help Long figure out $\boldsymbol{P}$ and $\boldsymbol{E}$ given the values of $\boldsymbol{R}$ and $\boldsymbol{D}$.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets.

Each dataset contains two space separated positive integers $\boldsymbol{R}$ and $\boldsymbol{D}$ ($1 \le \boldsymbol{R}, \boldsymbol{D} \le 100$)

Output

For each data set, write in one line the values of $\boldsymbol{P}$ and $\boldsymbol{E}$ with precision to four decimal places.

Sample Input	Sample Output

2
1 2
2 1
	0.8372 1.2732
	0.3183 0.3183