

## Problem A. Collectors

Input file:              
Output file:             
Time limit:            2 seconds  
Memory limit:         256 MiB

Yesterday Taja visited the museum. The tour was long and interesting, but the room she liked most contained collections of the cubes of the ten famous collectors. One of the cubes has attracted her attention a lot, but she forgot, who has possessed it. Nevertheless she memorized how three visible faces looked like, as well as criteria of each of the collector. You are to deduce the name of the collectors, who could possess such a cube, from this information.

Each cube has 6 faces. Each face has a single number written on it from 1 to 6, unique for each face. Numbers can be represented either by dots, or as a decimal or as roman numeral. Also each face has one of the following colors — Black, White, Green, Yellow, Skyblue, Red, Orange and Purple.

Here is the list of the names of the collectors and their corresponding criteria, which hold for the entire collection:

John	All numbers are represented as dots
David	Numbers are never written as roman numerals
Peter	All faces are white
Robert	Faces of the cube are either black or white
Mark	Odd numbers have white background, even numbers have black background
Paul	All prime numbers are written as decimal, and vice versa
Patrick	All face have the same color, but neither black nor white
Jack	All roman numerals are on the yellow background
Max	All faces have unique colors
Alex	Numbers of the same format have the same background, different formats have different colors

### Input

Input contains three lines, describing visible faces of the cube.

First symbol of  $i$ th line  $c_i$  ( $c_i \in \{B, W, G, Y, S, R, O, P\}$ ) — the color of  $i$ th face (Black, White, Green, Yellow, Skyblue, Red, Orange and Purple correspondingly). The, separated by space, follows the number written on the face, in one of the following formats:

1. From 1 to 6 «.» symbols (ASCII 46), which means that number is written with dots and the number equals to the number of this dots;
2. Decimal number from 1 to 6;
3. Roman numeral, written with «I» (ASCII 73) and «V» (ASCII 86) symbols.

It is guaranteed, that presented cube belongs to at least one collector.

### Output

Output should contain a single line with name of collectors, which can own this given cube. Names should be written in any order and separated by space.

All names of the collectors should be from the following list: John, David, Peter, Robert, Mark, Paul, Patrick, Jack, Max, Alex.

## Examples

input.txt	output.txt
W .. W ... W ....	John David Peter Robert Jack Alex
B 2 W 3 B 6	David Robert Mark Jack
G 1 G 2 G V	Patrick
G 2 G 3 Y ....	David Paul Jack Alex
W . B 2 W III	Robert Mark

## Problem B. Perfect gift

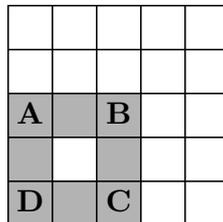
Input file: `input.txt`  
Output file: `output.txt`  
Time limit: 2 seconds  
Memory limit: 256 MiB

Taja prepares a present for the birthday. As you might know, the best present is the one handcrafted by yourself. Recently she learnt cross-stitching and decided to make use of this skill.

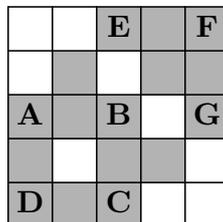
At home she only managed to find a canvas, which already had two crosses stitched on it. Don't panic — you can always complement it to the full picture. She had little experience, that's why she chose simple but nevertheless beautiful picture, which is parallelepiped. She wants to finish the present as soon as possible, thus number of new cross-stitches should be **the least possible**.

Parallelepiped on the **infinite** grid is drawn like this.

Let's draw a rectangle  $ABCD$  with its upper left corner at  $A$  and lower right corner at  $C$ .



Then draw segments of equal length towards up-right from  $A$ ,  $B$  and  $C$  — with ends at  $E$ ,  $F$ ,  $G$  correspondingly. Then add segments  $EF$  and  $FG$ .



All edges of the parallelepiped should be **at least 3 cells** long.

### Input

First line of the input contains two integers  $x_1$  and  $y_1$  — coordinates of the first cross-stitch. Second line contains coordinates of second cross:  $x_2$ ,  $y_2$ . Coordinates of the first cross-stitches are different. Axis  $OX$  is directed from left to right, and axis  $OY$  — from the bottom to the top. All numbers are within range  $[0, 10^9]$ .

### Output

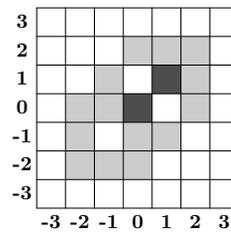
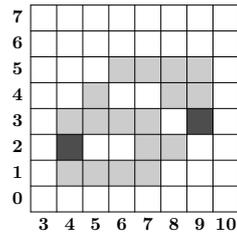
Output should contain single number — **the least** amount of required cross-stitches.

## Examples

input.txt	output.txt
4 2 9 3	17
0 0 1 1	14

## Explanation

This pictures correspond to the samples:



## Problem C. Casino

Input file:            `input.txt`  
Output file:          `output.txt`  
Time limit:           2 seconds  
Memory limit:        256 MiB

When Taja runs out of money, she goes to the casino. Recently a new game appeared at the casino, and Taja wants to master it. Help her.

Two parties of the game are croupier and visitor of the casino. Croupier has a single regular  $k$ -faced dice, which has all integers from 1 to  $k$  written on its faces. Croupier starts the game with rolling the dice once. Shown number determines amount of points gained by croupier.

To win, visitor has to gain more points, than croupier did. For this there's suggested a choice out of  $n$  options. Each option is a pair: the dice and number of its allowed rolls. Each face of each dice has some number written on it. This dice is rolled required number of times, all shown numbers are summed up and this sum is exactly the points gained by a visitor.

But some faces, in addition to numbers, have bonus marks. If shown face has bonus mark, then corresponding amount of points is added to the total, and visitor get additional dice roll. All faces of the same dice are pairwise distinct, which means there's no two identical bonus faces and no two identical ordinary faces. Each dice has at least one face without bonus mark. For every dice, the probability of each of its face being shown is the same.

In this problem it is required that for each possible amount of croupier's points from 1 to  $k$  you determine visitor's rolling option number, which leads to the maximal probability to gain strictly greater points than croupier did.

### Input

First line of the input contains single integer  $n$  ( $2 \leq n \leq 10$ ) — number of dice rolling options.

Next  $n$  lines contain descriptions of options in the following format.

First number  $c_i$  ( $1 \leq c_i \leq 10$ ) — number of allowed rolls. Second number  $f_i$  ( $2 \leq f_i \leq 12$ ) — number of dice faces. Next  $f_i$  numbers  $v_{ij}$  — numbers written on the faces.  $v_{ij}$  is either simply a number from 1 to  $f_i$ , meaning amount of points, or it can have additional plus sign «+» (ASCII 43) in front of the number, which is the bonus mark. For every dice, its numbers without plus sign are unique, all numbers with plus sign are unique, and there's at least one face without bonus mark.

Last line contains single integer  $k$ , which always equals  $\max_{1 \leq i \leq n} (c_i \times f_i)$ .

### Output

Output should contain  $k$  lines, each of which contains single integer  $b_i$  — number of the best option, which will allow to win with maximal probability by gaining more than  $i$  points (this probability shouldn't deviate from the right answer more than  $10^{-9}$ ).

Dice are numbered from 1 in the order they are given in the input.

## Examples

input.txt	output.txt
3 3 4 1 2 3 4 2 6 1 2 3 4 5 6 1 12 1 2 3 4 5 6 7 8 9 10 11 12 12	2 1 1 1 1 1 1 3 3 3 3 2
2 1 4 1 2 +1 +2 1 6 1 +1 2 3 4 5 6	2 2 2 2 1 1

## Explanation

Answer for the first sample could contain 1 on the first line, and the last could be any from 1 to 3.

## Problem D. In the cube

Input file:            .txt  
Output file:           output.txt  
Time limit:            2 seconds  
Memory limit:         256 MiB

Taja likes to go to the cafe «In the cube» with her friends, since it has very convenient ordering system. To make an order, guest should walk to the automated stand and choose any dishes they likes. There are several such stands and they all are fixed at specific place inside the cafe.

In the cafe guests sit in front of tables, there are  $k$  tables.  $i$ th table cannot serve more than  $c_i$  persons. Uncomfortableness of the table position is the sum of the distances from this table to  $c_i$  automated stands closest to it.

Formally, cafe is the grid  $(0,0) - (100,100)$ . Each cell  $(x,y)$  ( $0 \leq x,y \leq 100$ ) can contain either single automated stand or single table or nothing.

The distance between cells  $(x_1, y_1)$  and  $(x_2, y_2)$  equals to  $|x_2 - x_1| + |y_2 - y_1|$ .

You are to arrange the tables in such a way, that total sum of uncomfortablenesses for all tables should be minimal.

### Input

First line of the input contains two integers  $n$  and  $k$  ( $1 \leq n \leq 18$ ,  $1 \leq k \leq 200$ ) — amount of automated stands and tables correspondingly.

Following  $n$  lines contain coordinates of  $i$ th stand: two integers  $x_i$  and  $y_i$  ( $0 \leq x_i, y_i \leq 100$ ).

Next of each  $k$  lines contain single integer  $c_j$  ( $1 \leq c_j \leq n$ ) — number of seats at  $j$ th table.

### Output

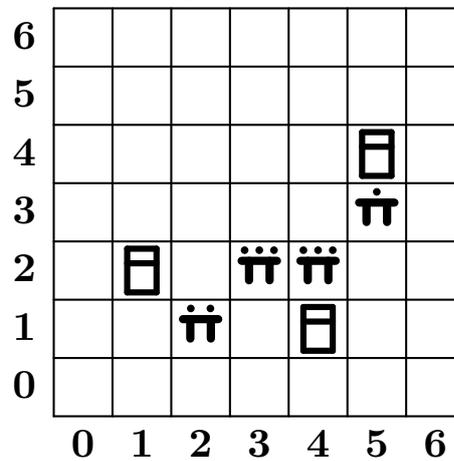
Output should contain single integer — minimal total uncomfortableness.

## Examples

input.txt	output.txt
<pre>3 4 1 2 4 1 5 4 1 2 3 3</pre>	<p>20</p>
<pre>2 10 0 0 100 100 1 1 1 1 1 1 1 1 1 1 1 1 1</pre>	<p>16</p>

## Explanation

Possible arrangement of the tables for the first sample looks like this:



## Problem E. Toy store

Input file:            `input.txt`  
Output file:          `output.txt`  
Time limit:           2 seconds  
Memory limit:        256 MiB

Taja often went by the toy store and looked at electronic table near the store window which displayed two integers. Store window shows several kinds of toys, but numbers on the table may be out of sync with actual number of different kinds. It turned out that, not every toy from the store window can be bought, because they do not remove toys as soon as possible, but only after some time past the purchase. For different kinds of toys this time can be different.

There are  $n$  kinds of toys. For each kind of toy it is known that initial amount of toys is  $c_i$  and time is  $t_i$  minutes after the purchase, after which the toy is removed from the window. Each minute the following happens:

- removal of toys from the store window, that were purchased corresponding amount of minutes ago;
- the electronic table is updated;
- new customer comes and **necessarily buys some toy, that is remaining in stock.**

Taja has been always interested the meaning of the numbers on the electronic table and recently she found it out. Both numbers show how many kinds of toys can be bought in the store, but first one shows number of kinds, **possibly** in stock up to the current moment, and second one is number of kinds, that are in stock up to the current moment **for sure**. Also Taja is interested how much is this table informative for customers. That's why she needs a program, that will model behaviour of the customers and update the table.

Your task is: for each minute calculate electronic table numbers.

### Input

First line of the input contains single integer  $n$  ( $1 \leq n \leq 100$ ) — number of kinds of the toys.

Each of the following  $n$  lines contains two integers  $c_i$  and  $t_i$  ( $1 \leq c_i \leq 100$ ,  $1 \leq t_i \leq 100$ ) — number of toys of  $i$ th kind and time, after which the toy will be removed out of the store window after the purchase correspondingly.

Next string contains single integer  $k$  ( $1 \leq k \leq 100$ ) — number of customers.

Each fo the following  $k$  lines contains integer  $q_i$  and  $q_i$  integers  $p_1, p_2, \dots, p_{q_i}$  — number of toys, that were removed at  $i$ th minute and numbers of these toys.

### Output

Output should contain  $k$  lines, each of which contains two integers  $a_i$  and  $b_i$  — numbers on the electronic table at moment of the beginning of  $i$ th minute correspondingly.

## Examples

input.txt	output.txt
3	3 3
1 2	3 2
2 1	3 2
3 3	2 2
6	2 2
0	1 1
0	
0	
3 1 2 3	
0	
1 2	

## Explanation

In the above example, the store window contains one toy of the first kind, two toys of the second kind and three toys of the third kind, which were removed after 2, 1 and 3 minutes correspondingly after the purchase. Numbers on the table should change in the following order:

- 3/3: there were no customers before the first one, he can buy any toy.
- 3/2: first customer could possibly buy the toy of first kind, thus there's no confidence, that second customer can buy it.
- 3/2: since the toy of neither the first kind nor second kind has been removed from the store window, then it means that first customer purchased the toy of the third kind. What was purchased by the second customer is impossible to decide yet.
- 2/2: no toys of the first kind anymore.
- 2/2: no toy has been removed from the store window, which means that previous customer bought a toy of the third kind.
- 1/1: There's only one toy of the third kind remaining.

## Problem F. Game with dices

Input file:            .txt  
Output file:           output.txt  
Time limit:            2 seconds  
Memory limit:         256 MiB

At her loft, Taja found an ancient tabletop game, which she managed to win only from time to time. Show Taja, how to win this game with a guarantee.

Game equipment consists of round piece with a radius of 1, which has an arrow drawn on top of it. Before starting to play one should mark a point on a table. The goal is to cover the marked point with the piece. It should be achieved by making moves with the following rule. First, player rotates the piece counterclockwise by integer amount of degrees from 0 to 359. Then the piece moves towards the arrow by the distance of 10.

Coordinates of the marked point is always  $(0, 0)$ .

Find any sequence of no more than 7500 moves, which solves the game.

### Input

First line contains two integers  $x, y$  ( $2 \leq \max(|x|, |y|) \leq 500$ ) — the initial position of the piece center.

Next line contains two real numbers  $vx, vy$  with 9 decimal places ( $vx^2 + vy^2 = 100$ ) — direction of the arrow.

### Output

First line must contain one integer number  $k$  — the number of moves ( $1 \leq k \leq 7500$ ).

Next line must contain  $k$  integers — rotation angles on the appropriate moves.

### Examples

input.txt	output.txt
10 -10	2
0.000000000 -10.000000000	180 90

## Problem G. Data Mining

Input file:              
Output file:              
Time limit:            2 seconds  
Memory limit:         256 mebibytes

One big IT-company use a data mining software, which was created without any paralellism. Employees of the company are starting this software time from time and wait till results will be calculated, spending man-hours of their working time.

The CEO of this company considers to rewrite this software using parallel technologies. To check if it will be really effective, CEO estimates the number of times this program will be executed within the next year (when the program will probably be rewritten).

On the other side, CEO knows development time for the parallel version of the program. And finally he can estimate execution time of both the parallel and the serial versions.

CEO wants to know in which cases parallel version gives the an advantage for the company in the man-hours against the sequential (for employees waiting for the results together with programmers rewriting the software).

### Input

First line of the input contains one integer  $T$  — number of test cases ( $1 \leq T \leq 1000$ ).

Then test cases follow, each on the separate line. Each test case consists of four integers  $d$ ,  $n$ ,  $p$  and  $s$  — development time, number of runs, time for one employee to wait for the parallel version results, time for one employee to wait for the serial version results, respectively ( $0 \leq d \leq 10^6$ ,  $0 \leq n \leq 10^5$ ,  $0 \leq s, p \leq 1000$ ).

### Output

For each test case print “**Rewrite**”, if rewriting the software will give the advantage in man-hours, “**Keep**”, if rewriting the software will cause more man-hours spent totally, or “**Flip a Coin**” if number of man-hours does not depend on CEO’s decision.

### Example

input.txt	output.txt
3	Keep
11 3 3 2	Rewrite
19 6 9 3	Flip a Coin
0 5 100 100	

## Problem H. Performance

Input file:            .txt  
Output file:           output.txt  
Time limit:            2 seconds  
Memory limit:         256 mebibytes

The performance of an multi-gear engine in a certain gear is not constant: it depends on the RPM of the engine. This relationship can be described using a performance curve.

For the “Parabolica” line of engines, the performance curve of all gears in the engine is a parabola of the form  $P = -aR^2 + bR + c$ , where  $R$  is the RPM of the engine, and  $P$  is the resulting performance.

Given the parabolas describing all gears in an engine, determine the gear with the highest maximum of performance.

It is guaranteed that answer is unique and the its maximum performance is at least 1 higher than the maximum performance in all the other gears.

### Input

First line of the input file contains integer  $T$  ( $1 \leq T \leq 100$ ) — number of test cases.

Each test case starts with line containing single integer  $n$  ( $1 \leq n \leq 10$ ): the number of gears in the engine. Then  $n$  lines follow, each containing three integers  $a$ ,  $b$  and  $c$  ( $1 \leq a, b, c \leq 10^4$ ) — parameters of the parabola, describing performance curve. Gears are numbered from 1 in order they given in the input file.

### Output

For each test case print one integer — number of gear with highest maximum performance.

### Examples

input.txt	output.txt
1	2
2	
3 130 1423	
2 153 210	

## Problem I. Tic-tac-toe

Input file:            Output file:           output .txt  
Time limit:            2 seconds  
Memory limit:         256 mebibytes

Given a partially played  $3 \times 3$  tic-tac-toe configuration after 7 turns, write a program to determine which player will have a better chance of winning if the game will be continued and the players will pick an empty square randomly.

### Input

Input file consists of the board in 3 consecutive lines, each consisting of exactly 3 lowercase characters (can be used only 'o', 'x' or '.' for empty square). There are exactly two empty squares left in the board. It is guaranteed that board is correct, i.e. difference between numbers of 'x' and 'o' is equal to 1 by absolute value and that game is not ended now (i.e. no three characters of same type are in some row, column or diagonal).

### Output

Print 'o' if player 'o' has a higher chance of winning, 'x' if player 'x' has a higher chance of winning, or "tie" otherwise.

### Examples

input .txt	output .txt
xo. oxo xx.	x

## Problem J. Move the Boxes

Input file:            Output file:           output .txt  
Time limit:            2 seconds  
Memory limit:         256 megabytes

There are  $m + 1$  positions located along a straight line, numbered by integers from 0 to  $m$  inclusive. The task is to move all the boxes from one position to another. You are allowed to use  $n$  cranes. Initially, all the cranes are in position 0. Each second, each crane can either stay on its position or move to an adjacent position to the left or to the right. Each crane can either be empty or hold a single box. The time required to attach the box to the crane or to detach it is negligible.

Find the least possible time required to move all  $k$  boxes from position 0 to position  $m$ . Each position can contain any number of boxes. Neither cranes nor boxes interfere with each other when moving.

### Input

The only line of input contains three integers  $n$ ,  $m$  and  $k$  ( $1 \leq n, m, k \leq 10^5$ ).

### Output

Output a single integer: the number of seconds required to move all  $k$  boxes from position 0 to position  $m$ .

### Examples

input .txt	output .txt
1 10 5	90
5 10 5	10