Information

Memory limit

The limit is 512 MiB for each problem.

Source code limit

The size of each solution source code can't exceed 256 KiB.

Submissions limit

You can submit at most 50 solutions for each problem.

You can submit a solution to each task at most once per 30 seconds. This restriction does not apply in the last 15 minutes of the contest round.

Scoring

Each problem consists of several subtasks. The subtask score is awarded if all tests in the subtask are passed.

The number of points scored for the problem is the total number of points scored on each of its subtasks. The score for the subtask is the maximum number of points earned for this subtask among all the solutions submitted.

Feedback

To get feedback for your solution, go to "Runs" tab in PCMS2 Web Client and use "View Feedback" link. In each problem of the contest you will see the score for each subtask, or the verdict for the first failed test.

Scoreboard

The contestants' scoreboard is available during the contest. Use "Monitor" link in PCMS2 Web Client to access the scoreboard. The standings provided in PCMS2 Web Client are not final.

Problem A. Expression Formatting

Time limit: 1 second
Memory limit: 512 megabytes

Masha is learning programming, but her teacher always says that she is doing wrong, and her code is not pretty enough. For instance, Masha doesn't like spaces, and she never uses spaces inside expression, for example: `(a+b)*c`.

The teacher tries to make Masha put the spaces around binary operators, for the expression to look like: `(a + b) * c`. Masha doesn't want to waste time on such stupid things, so she asks you to write a program that is putting spaces for her.

Input

The first line contains the expression that needs to be fixed. The expression consists of variable names, binary operators: '+', '-', '*', '/', and parentheses: '(', ')'. The expression doesn't contain any space, and is correct. All variables consist of a single lowercase English letter. The length of the expression string doesn't exceed 200.

Output

Print the same expression, adding a single space before and after each binary operator.

Scoring

Testing data for this problem consists of 20 test cases. For solving each test case you are awarded 5 points. Total score is the total sum of points for all test cases. The testing result for each test case is shown.

Examples

standard input	standard output
`a+b`	`a + b`
`((a))-b+(c*(d))`	`((a)) - b + (c * (d))`
`(a)/(b-b)+((d)+((c)))`	`(a) / (b - b) + ((d) + ((c)))`

Problem B. Contest Rescheduling

Time limit:	1 second
Memory limit:	512 megabytes

There are many programming competitions nowadays. For contestants' convenience competition organizers try to do their best to avoid time intersections for the competitions, but sometimes this happens.

Once two competitions were scheduled for the same day. The first competition was scheduled to start at s_1 with the duration of d_1 (ending at $s_1 + d_1$), and the second one at s_2 with the duration of d_2 (ending at $s_2 + d_2$). The duration of the competitions can't be changed, but the start time can. But there are restrictions for each competition: the first one shouldn't start before l_1, and end after r_1, the second one shouldn't start before l_2, and end after r_2.

You task is to write a program that can help organizers to reschedule the competitions, so that they don't intersect by their time, and given restrictions are still satisfied. If there are several new schedules, you are required to find one that the total change is minimized. Formally speaking, if c_1 and c_2 are the start times in the new schedule, then $(|s_1 - c_1| + |s_2 - c_2|)$ has to be minimized.

Input

Input data consists of several testcases.

First line contains an integer n — the number of testcases to solve ($1 \le n \le 50\,000$).

The description of n tests follow, each of them consists of two lines.

The first of these lines contains four integers l_1, r_1, l_2, and r_2 — the competitions' schedule restrictions ($0 \le l_i < r_i \le 10^9$).

The second one contains four integers s_1, d_1, s_2, and d_2 — the starting time and duration of each competition ($l_i \le s_i$; $s_i + d_i \le r_i$; $d_i \ge 1$).

Output

Pring n lines: the i-th line containing an answer for i-th testcase.

Print two integers c_1 and c_2 for each testcase: the starting time of the first and the second competition in a new schedule, respectively. If there are several solutions that minimize the total change, print any. If there is no way to make a new schedule, print two -1.

Scoring

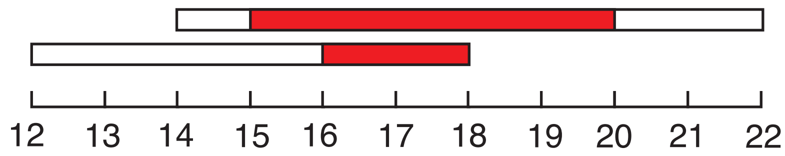
Subtask	Score	n	s_i, d_i	l_i, r_i
1	19	$n \le 1000$	$s_i, d_i \le 20$	$l_i = 0; r_i = 50$
2	20	$n \le 1000$	$s_i, d_i \le 50$	$l_i, r_i \le 50$
3	21	$n \le 1000$	$s_i, d_i \le 10\,000$	$l_i, r_i \le 10\,000$
4	15	$n \le 1000$	$s_i, d_i \le 10^9$	$l_i, r_i \le 10^9$
5	25	$n \le 50\,000$	$s_i, d_i \le 10^9$	$l_i, r_i \le 10^9$

Example

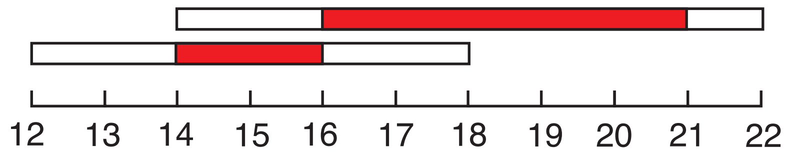
standard input	standard output
3 14 22 12 18 15 5 16 2 12 22 14 20 14 5 15 4 12 14 16 18 12 2 16 2	16 14 -1 -1 12 16

Explanation

Picture for the first testcase. Initial schedule:



New schedule:



The total change is $|15 - 16| + |16 - 14| = 3$. This is the minimum possible change.

Problem C. Advertisement Profit

Time limit: 1 second
Memory limit: 512 megabytes

Pashka is a popular vlogger. Currently there are 10 000 subscribers on his channel. He decided to plan next month releases.

Videos come in two types: regular and commercial. Pashka has n regular videos and m commercial videos. He can release the videos in any order he likes. Since Pashka is a top content maker, after the release of the i-th regular video, the number of subscribers on his channel will increase by a positive integer a_i. However, after the release of the i-th commercial video, the number of subscribers on its channel will decrease by a positive integer b_i. At the same time, if Pashka releases the i-th commercial video, and at that moment there will be s subscribers on his channel, the company that ordered this commercial will pay him $s \cdot c_i$ cents. It is guaranteed that even if Pashka releases all planned commercial videos and only them, the number of subscribers on the channel will remain nonnegative.

Pashka has not yet decided how many videos to release next month: he has q options. Each option is characterized by a single integer d_i — the number of videos that he wants to release. Now he asks you for each option to determine what maximum profit he can get.

Input

The first line contains a single integer n — the number of regular videos ($0 \le n \le 100$). Next n lines contain a single integer a_i — the change in the number of subscribers after the release of the i-th regular video ($1 \le a_i \le 100$).

Next line contains a single integer m — the number of prepared commercial videos ($0 \le m \le 100$). Next m lines contain two integers c_i and b_i — the number of cents that the company will pay for each subscriber at the time of the release of the video, and the change in the number of subscribers after the release of the i-th commercial video ($1 \le c_i, b_i \le 100$).

Next line contains a single integer q — the number of different options that Pashka wants to check ($1 \le q \le n + m + 1$). Next q lines contain a single integer d_i — the number of videos Pashka wants to release next month ($0 \le d_i \le n + m$). It is guaranteed that all d_i are different.

Output

For each option, output a single number — the maximum profit in cents that Pashka can get if he releases exactly d_i clips.

Scoring

Subtask	Score	Constraints			
		n	m	q	Additional
1	10	$n \le 5$	$m \le 5$	$q \le n+m+1$	—
2	11	$n \le 100$	$m \le 5$	$q \le n+m+1$	—
3	16	$n \le 100$	$m \le 100$	$q = 1$	$d_1 = n+m$
4	17	$n = 0$	$m \le 100$	$q \le n+m+1$	—
5	19	$n \le 100$	$m \le 30$	$q \le n+m+1$	$c_i, b_i \le 10$
6	27	$n \le 100$	$m \le 100$	$q \le n+m+1$	—

Examples

standard input	standard output
1 10 2 10 20 20 10 4 0 1 2 3	0 200000 299900 300200
3 10 40 30 3 5 10 15 20 1 100 7 0 1 2 3 4 5 6	0 150000 199900 209870 210710 211340 211550

Explanation

In the first example, if Pashka releases only one video, he should release the second commercial video, and then he will earn $10\,000 \cdot 20 = 200\,000$. If he wants to release two videos, then first he should release the second commercial video, and then the first commercial video. In this case, his profit will be equal to $10\,000 \cdot 20 + 9\,990 \cdot 10 = 299\,900$. If he wants to release all three videos, then first he should release a regular video, then the second commercial video, and then the first commercial video. The profit will be equal to $10\,010 \cdot 20 + 10\,000 \cdot 10 = 300\,200$.

Problem D. RSA factoring

Time limit: 1 second
Memory limit: 512 megabytes

RSA is a cryptosystem, which security relies on numbers $n = pq$, where p and q are different prime numbers. The number n is called an RSA modulus and is being used for calculations. RSA is believed to be secure because of the fact that for a given number n there is no known fast algorithm for factoring n into prime factors, when the bit length of n is high enough (1024 bits and longer). It is usually adviced to pick p and q as large random primes of approximately the same length. Generating the RSA modulus n is a process that doesn't allow mistakes, since there are many RSA attacks exploiting the faulty generation method. There are more details about RSA, but you are not going to need them to solve this problem.

When Carl read about using close primes in RSA, he implemented the following algorithm.

1. Generate a random prime number p_1 with b bits.
2. Starting from $p_1 + 1$, try all possible numbers in increasing order, until we encounter the next prime number p_2.
3. Return $n = p_1 p_2$.

Because on the first step we chose a random prime p_1 and the distance between adjacent primes is small on average, this algorithm is going to find p_2 rather quickly. Carl's friend Pierre realized that numbers found by Carl's algorithm can't be used for RSA, because it can be factored efficiently. Then Pierre offered to use not just two prime numbers, but four! Independently of p_1 we also choose a random b-bit prime number q_1 and the next prime after it q_2. Then, we set the RSA modulus to be equal to the product of all four primes: $n = p_1 p_2 q_1 q_2$. This algorithm turned out to be faulty too: it is still possible to factor such n.

You are given n generated using the first Carl's algorithm with 2 prime divisors, or with Pierre's updated algorithm with 4 prime divisors. Find all its prime divisors.

Input

The first line contains two integers b and k ($4 \le b \le 60$, $k = 2$ or $k = 4$). The next line contains number n in hexadecimal format, most significant digits first, without leading zeros.

It's guaranteed that n is a product of exactly k prime numbers, and it's generated randomly using either Carl's or Pierre's algorithm described in the problem statement. Each prime number has exactly b bits in binary, all prime divisors are different.

Output

Output k prime divisors of n in hexadecimal format without leading zeros, each one in its own line.

Scoring

Subtask	Score	Constraints
1	10	$b \le 8$, $k = 2$
2	10	$b \le 8$, $k = 4$
3	7	$b \le 15$, $k = 2$
4	8	$b \le 15$, $k = 4$
5	15	$b \le 30$, $k = 2$
6	15	$b \le 30$, $k = 4$
7	15	$b \le 60$, $k = 2$
8	20	$b \le 60$, $k = 4$

Examples

standard input	standard output
4 2 8f	b d
6 4 534ee3	25 29 3b 3d

Explanations

In the first example $n = \texttt{8f}_{16} = 143 = 11 \cdot 13$. $\texttt{b}_{16}$ is 11, and $\texttt{d}_{16}$ is 13. In the second example $n = 5459683$, which is represented as a product $37 \cdot 41 \cdot 59 \cdot 61$.

Problem E. Black Friday

Time limit: 1 second
Memory limit: 512 megabytes

Today is Black Friday, so Misha decided to purchase some new equipment for his mining farm. There is a special offer in the electronics store. If you buy video cards of type i, and the number of video cards is from l_i to r_i, inclusive ($1.4 \cdot l_i \le r_i$), then the price will be much lower than the market price. Misha can take with him no more than s video cards. He decided to buy as many video cards for this discount as possible. Misha can't solve this problem, so he asks you for help. For each type of video card, determine how many pieces you need to buy, or not to buy at all, so that the total number of video cards he bought is **maximum possible**, but not greater than s.

Input

The first line of input contains two integers n and s ($1 \le n \le 10^5$, $1 \le s \le 10^{13}$) — the number of video card types and the maximum number of video cards that Misha can take with him.

The following n lines contain two integers l_i and r_i ($1 \le l_i \le 10^{13}$, $1.4 \cdot l_i \le r_i \le 10^{13}$) — the minimum and maximum number of video cards of type i, that can be bought at a discount.

Output

In the first line print the only integer w — the maximum number of video cards that can be bought at a discount ($0 \le w \le s$). In the second line, print n integers x_i. The i-th number is equal to the number of video cards of i-th type to buy. Note that either $x_i = 0$ or $l_i \le x_i \le r_i$, and also that the sum of x_i must be equal to w. If there are several solutions maximizing w, print any.

Scoring

Subtask	Score	Additional Constraints
1	10	$n \le 20$
2	14	$n \le 500$, $s \le 2 \cdot 10^5$
3	16	$n \le 500$
4	29	$n \le 5000$
5	31	no additional constraints

Example

standard input	standard output
3 20 1 2 10 17 11 16	19 2 17 0

Explanation

In the example, it is optimal to buy video cards of the first and second type. Note that you cannot buy video cards of the second and third type at the same time, since then you will need to buy at least $10 + 11 = 21$ a piece, which is more than 20.