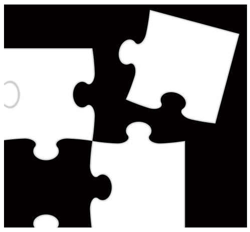
Croatian Open Competition in Informatics

Round 5, March 16th 2024

Tasks

Task	Time limit	Memory limit	Score
Zlagalica	1 second	512 MiB	50
Bitovi	1 second	512 MiB	70
Piratski kod	2 seconds	512 MiB	110
Rolete	2 seconds	512 MiB	110
Trokut	1 second	512 MiB	110
Total			450

Task Zlagalica



Little Maja has always loved puzzles. And since everyone knew that for a long time now, it is no wonder that one sunny day, Maja received an odd puzzle as a gift..
This puzzle has n pieces. Each piece has rectangular shape and is of a certain color. Also, each piece has 2 numbers written on its back: u and d. After a period of skillfully combining pieces and trying to fit them together, Maja figured out the meaning of those numbers.

She found out that number u represents "direction", in other words, does the next piece of the puzzle connect with the current one from the upper or from the right side of the current piece. Number d specifies the starting column/row where we connect the next piece of the puzzle with current one. In more detail:

- If u is equal to 0, we add next piece **above** the current one by connecting its **bottom left** corner with current piece's **top** edge at **column d**.
- If u is equal to 1, we add next piece to the **right** by connecting its **bottom left** corner with current piece's **right** edge at **row d**.

Let's demonstrate this for pieces colored in colors "a" and "b". *Picture 1* shows the case where $u = 0$, and $d = 3$. *Picture 2* shows case when $u = 1$ and $d = 3$. (In both cases, numbers u and d represent numbers written on the back of piece colored "a").

```
. . b b b b b
. . b b b b b
a a a a . . .
a a a a . . .
a a a a . . .
```

Picture 1

```
a a a a . . . . .
a a a a b b b b b
a a a a b b b b b
```

Picture 2

Maja has grown tired of this puzzling puzzle, but her curiosity knows no bounds! That's why she's asking for your help. She's interested in knowing, for a given description of every piece of the puzzle and the sequence of their placement, what will the completed puzzle look like? Write a program that prints the dimensions (height and width) of the completed puzzle, as well as its final appearance within a rectangle of the same height and width, where "." represents places where there is no part of the puzzle.

Input

In first row, there is n ($1 \le n \le 20$), number of puzzle pieces.

In the i-th of next n rows there are per 1 character and 4 integers, in order: b_i, r_i, s_i, u_i, d_i - description of i-th piece:

- b_i will always be 1 lowercase letter of english alphabet, and it represents the color of the i-th puzzle piece
- r_i and s_i ($1 \le r_i, s_i \le 10$) represent in order, number of rows and columns of i-th puzzle piece
- u_i and d_i ($0 \le u_i \le 1, 1 \le d_i \le r_i, s_i$ (depends on u_i)) are numbers on the back of i-th puzzle piece, same as in the task statement.

In the last row of input there are n integers, order in which pieces are connected, where number i ($1 \le i \le n$) represents i-th puzzle piece in input. Each puzzle piece will appear in the sequence exactly once.

Output

Print the height and width of the completed puzzle. After that, print the appearance of the puzzle within a rectangle of the same height and width. In the places within the rectangle where there is no part of the puzzle, print ".".

Scoring

Subtask	Points	Constraints
1	17	The order of connecting the puzzle pieces will be identical to the order of inputting them.
2	12	For each puzzle piece: $u = 0$.
3	12	For each puzzle piece: $u = 1$.
4	9	No additional constraints.

Examples

input

```
2
a 3 4 0 3
b 2 5 1 1
1 2
```

output

```
5 7
..bbbbb
..bbbbb
aaaa...
aaaa...
aaaa...
```

input

```
2
a 3 4 0 3
b 2 5 1 1
2 1
```

output

```
4 9
.....aaaa
.....aaaa
bbbbbaaaa
bbbbb....
```

input

```
4
g 9 5 0 2
a 3 2 1 1
c 5 10 0 2
p 8 7 1 6
4 3 2 1
```

output

```
18 17
..........ggggg..
..........ggggg..
..........ggggg..
..........ggggg..
..........ggggg..
..........ggggg..
..........ggggg..
..........ggggg..
........aaggggg..
........aa.......
ppppppp.aa.......
pppppppcccccccccc
pppppppcccccccccc
pppppppcccccccccc
pppppppcccccccccc
pppppppcccccccccc
ppppppp..........
ppppppp..........
```

Task Bitovi

What came first, the chicken or the egg? Is it better to live a hundred years as a millionaire or seven days in poverty? How to become a chess grandmaster? ~~How to raise blinds?~~ How to pass the final exams? How to train a dragon? These are interesting questions we can ponder only after the competition, but now we offer one *less interesting* computer science problem.

You are given two sets of numbers A and B of size N. In one move, you can select an arbitrary element from set A and change one arbitrary digit (bit) in its binary representation. The resulting number must not be an element of set A immediately before the change.

For example, the number 5 in binary is 0101_2. In one move, it can become $13 = 1101_2$, $1 = 0001_2$, $7 = 0111_2$, or $4 = 0100_2$ if we change its 4th, 3rd, 2nd, or 1st bit, respectively.

Determine a sequence of moves by which set A becomes equal to set B. Sets are equal if they have the same size and there is no element in set A that does not belong to set B.

Note: The number of moves does not have to be minimal, but it must satisfy the task constraints.

Input

The first line contains the integer N ($1 \le N \le 2^{15}$), the size of the sets A and B.

The second line contains N different integers a_i ($0 \le a_i < 2^{15}$), the elements of the set A.

The third line contains N different integers b_i ($0 \le b_i < 2^{15}$), the elements of the set B.

Output

In the first line, print the number of required moves.

In the remaining lines, print the numbers x and y ($0 \le x, y < 2^{15}$) – we change the number x from set A to the number y. The numbers x and y must differ by exactly one bit, and $x \in A$ and $y \notin A$ must hold at the moment we execute the move.

Scoring

The number of moves you are allowed to make in all subtasks must not exceed 2^{19}. It can be shown that a solution always exists within the given constraints.

Subtask	Points	Constraints
1	10	$a_i, b_i \le 2^{14}$
2	15	$N \le 7$
3	30	$N \le 2^7$
4	15	No additional constraints.

Examples

input	input	input

`3`
`0 1 2`
`1 2 3`
	`3`
	`4 8 31`
	`0 4 8`
		`5`
		`0 1 2 4 5`
		`7 6 5 3 2`

output	**output**	**output**

`2`
`1 3`
`0 1`
	`5`
	`31 30`
	`30 28`
	`28 24`
	`24 16`
	`16 0`
		`9`
		`1 3`
		`3 7`
		`0 1`
		`1 0`
		`2 6`
		`0 2`
		`7 3`
		`5 7`
		`4 5`

Clarification of the first example:
If we first make the move `0 1`, and then `1 3`, between these two moves, set A would have two identical elements, which the task does not allow. Another possible solution is `2 3`, `0 2`.

Clarification of the second example:
$31 = 11111_2$. By removing bits from least to most significant, we obtain the numbers 30, 28, 24, 16, and 0 in sequence. After all moves, set A becomes equal to set B.

Task Piratski kod

Captain Marrrtina, together with her pirate crew, after three months of searching for long lost treasure belonging to the most famous Italian pirate finally dug up chest full of treasure. But to unlock the chest she needs a secret combination which is described in a message in a bottle next to the chest.

The message says:

So that only the most worthy pirate shall be able to open the chest, the combination is the solution to the following puzzle: A binary sequence s of length a in which the only pair of consecutive ones is located at the end of the sequence is a pirate representation of a number x if

$$s[0] \cdot \mathrm{Fib}[2] + s[1] \cdot \mathrm{Fib}[3] + s[2] \cdot \mathrm{Fib}[4] + \ldots + s[a-2] \cdot \mathrm{Fib}[a] = \sum_{i=0}^{a-2} s[i] \cdot \mathrm{Fib}[2+i] = x,$$

where $\mathrm{Fib}[x]$ denotes the x-th Fibonacci number. Fibonacci numbers are defined as following: $\mathrm{Fib}[1] = 1, \mathrm{Fib}[2] = 1, \mathrm{Fib}[y] = \mathrm{Fib}[y-1] + \mathrm{Fib}[y-2]$ for each $y > 2$.

For example $11_p = 1, 011_p = 2, 1010011_p = 17$, where $_p$ denotes a pirate representation of a number.

A pirate code is a binary sequence (without any condition on consecutive ones) that represents a sequence of positive integers. To read it we partition it in as many parts as possible that are pirate representation of some numbers (and possibly a suffix that is not a pirate entry of any number) and write those integers in a sequence. For example we partition 01111010110101 in 011|11|01011|0101, the last part is not a pirate representation so we delete it 011|11|01011 and read a sequence $2, 1, 7$.

The value of a pirate code is equal to the sum of values of decoded sequnce of positive integers. The value of previous code is 10.

My favourite number P is the sum of values of all pirate codes of length k. As that number may large, the combination to the chest is the remainder of P modulo $10^9 + 7$

- Leonarrrdo da Pisa

If Marrrtina doesn't manage to open the chest, her crew will not consider her worthy and they'll make her walk the plank.

Input

In first and only line there is a positive integer $n \leq 5000$.

Output

In a single line of output print n numbers where k-th number represents the answer to the puzzles for codes of length k.

Scoring

Subtask	Points	Constraints
1	20	$n = 20$
2	40	$n = 300$
3	50	$n = 5000$

Examples

input

```
4
```

output

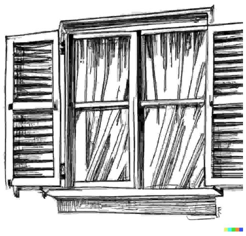
```
0 1 4 16
```

Clarification:
Codes of length 1 are 0 and 1 and they both have value 0.

Code 11 has value 1 while all other codes of length 1 have value 0.

Code 1111 has value 2, and code 0011 has value 3.

Task Rolete



One Saturday Luka woke up from an afternoon nap and remembered: today is COCI! There was only one thing that he needed to do before the contest: raise the blinds.

Luka has n blinds in his room, where the i-th one is lowered by a_i centimeters from the top of the window. He can raise the blinds in two ways:

- He can start lifting any singular blind manually. With this method, it takes him t seconds to raise the blind by 1 centimeter.
- He can press a button, which starts raising all blinds parallel at the same speed.

The speed at which the blinds are raised with a button is defined as follows: If all blinds are still rising, each will rise by 1 centimeter in s seconds. If r blinds have already been risen to the top that slows down the system. Then it will take $s + k \cdot r$ seconds for all the remaining blinds to rise by 1 centimeter.

COCI is about to start, and Luka wants to raise his blinds as soon as possible. Meanwhile, his brother Marin entered the room and asked him q questions: *What is the minimum time you need to raise the blinds so that they are all lowered by at most h centimeters?* Marin is interested in the answer for each question considering the initial state of the blinds.

They realized that there is not enough time to think about it before COCI. Fortunately, the problem has just appeared here as well! Help them solve it!

Note: Luka will always raise the blind by an integer value of centimeters.

Input

The first line contains integers n, t, s and k ($1 \le n, t, s \le 10^5, 0 \le k \le 10^5$), the number of blinds, the time required to raise a blind manually, the time required to raise a blind with a button and the slowing factor of parallel raising.

The second line contains n integers a_i ($0 \le a_i \le 10^5$), the initial state of blinds.

The third line contains integer q ($1 \le q \le 10^5$), number of questions.

The fourth line contains q integers h_i ($0 \le h_i \le 10^5$), required maximal blind height.

Output

In first and only line print q numbers, i-th of them is minimum time for raising the blinds such that they are lowered by at most h_i centimeters.

Scoring

Subtask	Points	Constraints
1	16	$n, q, a_i, h_i \le 100$
2	26	$k = 0$
3	32	$q = 1$
4	36	No additional constraints.

Examples

input	input	input
3 2 5 1 2 2 4 3 2 0 1	2 3 4 0 3 1 3 3 2 0	4 3 10 3 2 4 5 6 3 4 3 0
output	**output**	**output**
4 14 9	0 3 10	9 18 47

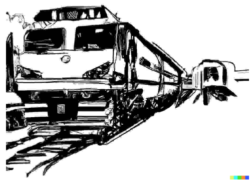
Clarification of the first example:

To have all blinds lowered by at most 2 centimeters, Luka needs to manually raise the third blind by 2 centimeters. The quickest way to do this is to raise it manually. This will take him 4 seconds.

If all blinds need to be fully raised, Luka can first raise the third blind by 2 centimeters manually. Now he can press the button and let the blinds rise in parallel by 2 centimeters. In total, that will take $4 + 10 = 14$ seconds.

Similarly, if the blinds need to be lowered by at most 1 centimeter, Luka will first raise the third blind by 2 centimeters, and then raise all blinds together by 1 centimeter. The total time for raising will be 9 seconds.

Task Trokut



Ivan and Lucija are on a journey to a place far... far away. They know that the journey will last a long time and that they will be bored at some point. While they were thinking about what to do, Lucija came up with a game.

She drew N points on paper so that they form the vertices of a regular N-gon and labeled them sequentially from 1 to N. The player whose turn it is selects two of the drawn N points such that the line segment connecting those two points does not intersect any of the previously drawn line segments and connects those two points. Line segments are **allowed** to touch at vertices. A player wins if after their move there exist three connected line segments forming a triangle, i.e., if there exist three points such that they are all connected by the drawn line segments. Of course, players are allowed to connect adjacent vertices, and those line segments are considered for triangle formation. Players take turns, and Lucija is the first to play.

Both are extremely skilled players, and we know they will play optimally. Your task is to determine, for a given N, who will be the winner of the game. It can be shown that the game will always end after a finite number of moves and that there will always be a winner.

Input

In the first row there is an integer T ($1 \le T \le 10\,000$), number of scenarios. In the next T rows there is an integer N ($3 \le N \le 10^9$), number of points Lucija drew on a piece of paper.

Output

In T rows, for every scenario in given order, output `"Ivan"` or `"Lucija"` (without quotes), winner in a given scenario.

Scoring

Subtask	Points	Constraints
1	13	$T \le 18, N \le 20$
2	36	$T \le 998, N \le 1\,000$
3	15	$N \le 10^5$
4	46	No additional constraints.

Examples

input

```
3
3
4
5
```

output

```
Lucija
Lucija
Ivan
```

input

```
3
7
8
9
```

output

```
Lucija
Lucija
Ivan
```

Clarification of the first example:

When $N = 3$, then all three possible line segments must be connected, and Lucija wins. When $N = 4$, then Lucija can connect the line segment between points 1 and 3. We see that after any move by Ivan, Lucija can connect a triangle and win.