

## Problem A. Count The Ones

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

Given three integers  $a$ ,  $b$  and  $c$ . Calculate number of 1's in the binary representation of number  $2^a + 2^b - 2^c$ .

### Input

Input consists of three integers  $a$ ,  $b$  and  $c$  ( $1 \leq c < b < a \leq 10^9$ ).

### Output

Print one integer — number of 1's in the binary representation of  $2^a + 2^b - 2^c$ .

### Example

standard input	standard output
3 2 1	2

## Problem B. Craters

Input file:            *standard input*  
Output file:           *standard output*  
Time limit:            1 second  
Memory limit:         256 mebibytes

Craters on the moon can be represented as  $N$  randomly chosen integer points on the plane.

You control the rover; the rover is going to place three sensors in some craters.

To do it effectively, you must find out the triangle with the vertices in crater points and maximum possible area.

### Input

First line of the input contains one integer  $N$  — number of points ( $3 \leq N \leq 2 \times 10^5$ ).

Each of next  $N$  lines contain two integers — coordinates of the next point  $x_i$  and  $y_i$ .

It is guaranteed that each of numbers  $x_i$  and  $y_i$  was chosen equiprobably from  $2N + 1$  integers between  $-N$  and  $N$ .

### Output

Print three lines, each line must contain coordinates of the next vertice of the triangle. If there are multiple solutions, print any of them.

### Example

standard input	standard output
5	4 -3
1 5	5 3
-2 5	-2 5
-1 2	
4 -3	
5 3	

## Problem C. Closing Ceremony

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

In the Berland, Qualification round of ACM ICPC Subregional Contest is finished. Director of the biggest site plans to announce the winning team at the closing ceremony. Unfortunately the Jury was calculating quotas for the final round of Subregional Contest, so they have just emailed contest log to him.

The header of the contest log contains a list of all teams with the number of solved problems and penalty times, ordered by some internal parameter. Following the ACM ICPC rules, the team which solved more problems, than all other teams is declared a winner.

In case of a tie between several teams, team with least penalty time between those teams is declared a winner.

It is guaranteed that no two teams have same number of solved problems and penalty time.

Director asked you to write a program, which reads the log and prints the name of the winning team.

### Input

First line of the input contains one integer  $n$  ( $n \leq 100$ ) – number of teams participating in Qualification Round. Then  $2n$  lines follow – team descriptors. Each team is described in two sequential lines. First line of the team description contains team name (nonempty string, consisting of no more than 20 uppercase or lowercase English letters, other characters are not allowed). It is guaranteed that team names are pairwise distinct. Second line of the team description contains two integers  $p_i$  ( $0 \leq p_i \leq 15$ ) and  $t_i$  ( $0 \leq t_i \leq 5000$ ) – number of solved problems and penalty time, respectively. Order of teams in the input file usually is different, than order in the standings table.

### Output

Print the line containing one string – name of the winning team.

### Example

standard input	standard output
5 Harvard 10 1358 MIPT 10 1437 Shanghai 11 1567 SPbSU 11 1560 Warsaw 10 1586	SPbSU

## Problem D. Skyscrapers

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

At the main street of the city Yecho there are  $N$  skyscrapers, standing sequentially one next to other. If look leftside right, sequence of their height is  $a_1, a_2, \dots, a_N$ .

Once Ugurbato, the sea monster, invaded this city. When Ugurbato hits the skyscraper, it is destroyed; moreover, Ugurbato is so strong that he creates a blast wave which is going to the left and to the right side from the skyscraper hit by monster.

Consider the left part of the blast wave. Let's say the skyscraper  $i$  was hit by Ugurbato. If blast wave meets already destroyed skyscraper, it decays. If blast wave meets non-destroyed skyscraper with number  $k$ , it is destroyed if and only if  $a_i - a_k \geq |i - k|$  and blast wave does not decay at this skyscraper regardless of its destruction. For the right part of the blast wave process is similar.

Ugurbato hits  $M$  times, and each next hit is done when both blast waves decay.

Print the number of skyscrapers which are destroyed after each Ugurbato's hit.

### Input

First line of the input contains one integer  $N$  ( $1 \leq N \leq 10^5$ ) — number of the skyscrapers.

Second line contains  $N$  space-separated integers  $a_1, a_2, \dots, a_N$  — heights of the skyscrapes, listed leftside right ( $1 \leq a_i \leq 10^9$ ).

Third line contains one integer  $M$  ( $1 \leq M \leq N$ ) — number of Ugurbato's hits.

Fourth line contains  $M$  space-separated pairwise distinct integers  $b_1, b_2, \dots, b_M$  — nunmbers of the skyscrapers, which were hit by Ugurbato, in order (skyscrapers are numbered from left to right, starting from one). It is guaranteed that for any  $i$  skyscraper  $b_i$  was not destroyed by the blast wave before it was destroyed directly by Ugurbato's hit.

### Output

Print  $M$  lines, each containing one integer — number of skyscrapers, destroyed after  $i$ 'th hit of the monster.

### Examples

standard input	standard output
5 3 2 4 10 5 2 3 4	2 2
5 1 2 3 2 1 1 3	5

## Problem E. Blackboard

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

At the math lesson teacher explained several regular ways to fill matrix  $N \times N$  with sequential integers from 1 to  $N^2$ . First, he draws the empty matrix  $N \times N$  (at the figure below  $N = 3$ ).


Then he filled matrix with the sequential integers in four different ways, starting from leftmost upper cell:

(1)	<table border="1" style="display: inline-table;"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>6</td><td>5</td><td>4</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	6	5	4	7	8	9	(2)	<table border="1" style="display: inline-table;"><tr><td>1</td><td>6</td><td>7</td></tr><tr><td>2</td><td>5</td><td>8</td></tr><tr><td>3</td><td>4</td><td>9</td></tr></table>	1	6	7	2	5	8	3	4	9	(3)	<table border="1" style="display: inline-table;"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>8</td><td>9</td><td>4</td></tr><tr><td>7</td><td>6</td><td>5</td></tr></table>	1	2	3	8	9	4	7	6	5	(4)	<table border="1" style="display: inline-table;"><tr><td>1</td><td>8</td><td>7</td></tr><tr><td>2</td><td>9</td><td>6</td></tr><tr><td>3</td><td>4</td><td>5</td></tr></table>	1	8	7	2	9	6	3	4	5
1	2	3																																									
6	5	4																																									
7	8	9																																									
1	6	7																																									
2	5	8																																									
3	4	9																																									
1	2	3																																									
8	9	4																																									
7	6	5																																									
1	8	7																																									
2	9	6																																									
3	4	5																																									

Then he asked Vasya to fill the matrix  $N \times N$  in similar ways. Vasya is too lazy to do it, so he asked you to write a program to do it.

### Input

Input consists of two integers  $N$  ( $1 \leq N \leq 100$ ) and  $a$  ( $1 \leq a \leq 4$ ), where  $a$  defines the way to fill the matrix.

### Output

Print  $N$  lines, each containing  $N$  space-separated integers — the resulting matrix.

### Examples

standard input	standard output
3 1	1 2 3 6 5 4 7 8 9
3 2	1 6 7 2 5 8 3 4 9
3 3	1 2 3 8 9 4 7 6 5
3 4	1 8 7 2 9 6 3 4 5

### Note

Samples coincide with four matrices, presented by teacher.

## Problem F. Buddy Numbers

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

Polycarp says that two positive integers are *buddies*, if one is divisible by another. For example, 2 and 4 are buddies, and 10 and 3 are not. Note that 1 is buddy with every positive integer.

Given integer  $n$  find out if it is possible to arrange all consecutive integers from 1 to  $n$  in a row such that any pair of neighbor numbers are buddies. If it is possible, print any of those arrangements.

### Input

The only line of the input contains one integer  $n$  ( $1 \leq n \leq 1000$ ).

### Output

If it is impossible to arrange numbers from 1 to  $n$  in such a way, print  $-1$ . Otherwise print one of possible arrangements as  $n$  space-separated integers in one line. If there are several possible arrangements, print any of them.

### Examples

standard input	standard output
2	2 1
3	3 1 2

## Problem G. Gmoogle

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 megabytes

You are hired to create alpha version of the new searching engine named GMoogle. Alpha version should work with the content, represented as a database of sentences:

- Content is merged into line  $S$ , consisting of characters ‘a’-‘z’, ‘A’-‘Z’, spaces, notation marks (“.!?”) (quotes are not counted) and decimal digits.
- If one of characters “.!?” presents in the  $S$ , then it denotes the end of the sentence, except for one special case: if first non-space character after ‘.’ is lowercase English letter, then it is an abbreviation sign but not the end of the sentence; for example, string “I like tea in a 500 ml. cup” contains one sentence, but strings “Cup is 500 ml. I want it” and “Cup is 500 ml. 500 ml is great for me” contains two sentences).
- First non-space character after the end of sentence is considered as the first character of the new sentence.
- *word* is contiguous sequence of characters ‘a’-‘z’, ‘A’-‘Z’, delimited by spaces, notation signs or beginning/end of the sentence/string. It is guaranteed that digits can not be neighbors of the letters, i.e. sequences like “10ml” or “R2D2” are illegal.
- $S$  may contain the sentences containing no words. It is guaranteed that  $S$  does not contains two or more characters “.!?” in a row.

After the content is indexed, users make requests. Each request can be represented as a string  $q$ , consisting of one or more words (definition of the word is given above). Words are separated by arbitrary number of spaces (1 or more), heading and trailing spaces are possible.

Your program has to print all sentences from  $S$ , where all words from  $q$  are presented (in any order). Words are considered equal, if all the letters at the corresponding positions are the same (case insensitive, i.e. ‘B’ and ‘b’ are considered the same).

### Input

First line of the input contains non-empty line  $S$ , consisting of no more than 1000 characters. Next line contains one integer  $n$  ( $1 \leq n \leq 100$ ) — number of the requests. Then  $n$  requests  $q_1, \dots, q_n$  follow, each on separate line in the format, described above. Note that in  $S$  and  $q_i$  trailing and heading spaces are allowed.

### Output

For each request  $q_1, q_2, \dots, q_n$  print the request at the separate line. Then print the list of found sentences in same order they present in  $S$ , one sentence per line. Requests and answers are printed in the quotes; answers are preceded by single ‘-’ and single space; heading and trailing spaces must be eliminated. Look the sample for clarify.

## Example

standard input	standard output
Hello everyone. I want 2 coffee if you have it. I like coffee very much. 4 HELLO Coffee much        coffee VoDka	Search results for "HELLO": - "Hello everyone." Search results for "Coffee": - "I want 2 coffee if you have it." - "I like coffee very much." Search results for "much        coffee": - "I like coffee very much." Search results for "VoDka":



## Problem H. Generator

Input file:            *standard input*  
Output file:           *standard output*  
Time limit:            1 second  
Memory limit:         256 mebibytes

Consider next algorithm, which generates random prime number between 2 and  $N$ :

1. Choose some random integer  $x$  from the range  $[2, N]$ . For all  $N - 1$  integers the probability to be selected is the same.
2. Check if  $x$  is prime. If yes, go to step 3, else go to step 1.
3. Return  $x$  as the result.

To check if  $x$  is prime, the following algorithm is used:

1. Let  $d := 2$ .
2. If  $d$  is greater than square root from  $x$ , i.e.  $d^2 > x$ , then terminate the process and tell that  $x$  is prime. Else go to step 3.
3. Check, if  $x$  is divisible by  $d$ . If yes, then terminate the process and tell that  $x$  is not a prime. Else let  $d := d + 1$  and go to step 2.

For a given  $N$ , find out expected numbers of divisions at step 3 in second algorithm.

### Input

First line contains one integer  $T$  ( $1 \leq T \leq 10^5$ ) — number of testcases.

Each testcase is placed on the separate line and consists of one integer  $N$  ( $2 \leq N \leq 10^7$ ).

### Output

For each test case, print the answer in the form of irreducible fraction.

### Example

standard input	standard output
6	0/1
2	0/1
3	1/2
4	2/3
5	1/1
6	2/1
10	

## Problem I. Addition

Input file: *standard input*  
Output file: *standard output*  
Time limit: 5 seconds  
Memory limit: 256 mebibytes

This is an interactive problem.

Polycarp is a software engineer in the «Abacus» company. Each day he must generate  $N$  binary strings of length  $M$ .

Once Polycarp get bored and decided to play the following game: when he generates next binary string, he treats it as a number written in binary representation (it is possible that number contains leading zeroes). Then he adds that number to each of previously generated strings (also treated as numbers) and counts the number of times when the sum can't be written with  $M$  or less binary digits.

But Polycarp is not so good in the binary addition, so he asks you to help him.

### Interaction Protocol

At the start of interaction process, the jury program sends to your program two positive integers  $N$  and  $M$  ( $1 \leq N \times M \leq 10^5$ ) — number of binary strings, generated by Polycarp, and requested length, appropriately.

Then next scenario repeats  $N$  times: the jury program sends at the new line to your program next binary string of length  $M$ , and your program must print at the new line one integer — number of times when this string, treated as integer in binary representation, added to one of previously generated strings is too big to fit  $M$  binary digits.

When all  $N$  strings are processed, your program must exit with exit code 0, else interaction will cause an error (even if all answers will be correct).

### Examples

standard input	standard output
5 2	0
01	0
10	2
11	0
00	2
10	
5 3	0
110	0
001	1
101	2
110	3
011	

### Note

Please do not forget to flush output buffer after each printed answer, i.e. do next:

- For Pascal: `flush(output);`
- For C/C++: `fflush(stdout)` or `cout.flush();`
- For Java: `System.out.flush();`
- For Python: `sys.stdout.flush()` from library `sys`;

- For C#: `Console.Out.Flush()`;

Also dont forget to print the newline character after each number you output.

## Problem J. White Mage

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

On the ancient battlefield the White mage fights army of  $n$  evil creatures. The mage can use  $m$  attack spells. Each spell can kill set of enemies, specific for that spell, and for casting this spell some mana will be used. Note that:

1. Spell can be used only if all creatures to be killed by this spell are alive before it was cast, otherwise it does nothing.
2. Once casted, spell destroys all related to it creatures, the mage cant control it to protect some of them.

Check if the White Mage can destroy all evil creatures and, if yes, how is minimum amount to mana needed for victory?

### Input

First line of the input consists two integers  $n$  ( $1 \leq n \leq 18$ ) – number of evil creatures and  $m$  ( $0 \leq m \leq 100$ ) – number of attack spells.

Each of next  $m$  lines defines one spell and contains list of affected creatures: integer  $k_i$  ( $1 \leq k_i \leq n$ ) – length of the list, then come  $k_i$  integers between 1 and  $n$ , inclusively – numbers of creatures, affected by this spell – номера существ, and last integer in the spell definition is  $v_i$  ( $v_i \leq 1000$ ) – amount of mana used to cast this spell.

### Output

Print one integer – minimal amount of mana, needed to destroy all evil creatures. If it is impossible to destroy them all, print  $-1$  instead.

### Examples

standard input	standard output
5 6 2 1 2 10 3 3 4 5 18 2 4 5 6 1 3 7 1 2 4 1 1 11	23
3 2 2 1 2 5 2 2 3 5	-1

### Note

In the first example mage have 4 ways to win: (spells 1 and 2 – 28 mana), (1,3,4 – 23), (2,5,6 – 33), (3,4,5,6 – 28). So using 1 3 4 is the most effective way.

In the first example creature with number 2 will be killed after casting any of the spells. So only one of spells can be casted, but it can't destroy all evil creatures.

## Problem K. Inspection

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

Polycarp visited Bytegorsk, one of cities of Berland, to inspect all the bus routes in the city.

The road network in Bytegorsk consists of  $N$  bus stops and several bidirectional roads, connecting them in such a way that between any two bus stops exists only one path (indirect or through other bus stops). Time for bus to pass distance between any two consecutive (i.e. directly joined by the road) cities is 1 minute.

For each two bus stops in Bytegorsk exists express-route of the bus, which connects those bus stops and is ignoring other stops on the way. Polycarp wants to inspect each of those routes atleast once (does not matter, in which direction).

Calculate the minimum time Polycarp need to inspect all express-routes.

### Input

First line of the input contains one integer  $N$  — number of bus stops ( $1 \leq N \leq 2 \times 10^5$ ).

Each of next lines contains description of the one road — two integers  $a$  and  $b$  ( $1 \leq a, b \leq N; a \neq b$ ) — number of stops it connect. It is guaranteed that each pair is listed no more than once.

### Output

Print one integer — minimum time Polycarp need to inspect all express-routes.

### Examples

standard input	standard output
5 1 2 1 3 3 4 3 5	18
2 2 1	1

## Problem L. Fibonacci Equation

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

Bison Mike chose three consecutive Fibonacci numbers:  $F_n$ ,  $F_{n+1}$  and  $F_{n+2}$ , shuffled them and used as  $A$ ,  $B$  and  $C$  in the quadratic equation

$$Ax^2 + Bx + C = 0$$

Now Mike wants to know the number of different real root of this equation. Help him to find out the answer.

### Input

First line of the input consists of three integers  $i, j$  and  $k$  ( $0 \leq i, j, k \leq 10^9$ ) — indices in the Fibonacci sequence; where  $A = F_i$ ,  $B = F_j$  and  $C = F_k$ . It is guaranteed that  $i, j$  and  $k$  are pairwise distinct and that difference between maximum and minimum of those integers is equal to 2.

### Output

Print one integer — number of different roots of the equation.

### Examples

standard input	standard output
1 2 0	2
1 0 2	0

### Note

Rules to build Fibonacci sequence:

$$F_0 = 0$$

$$F_1 = 1$$

$$F_i = F_{i-1} + F_{i-2}, \text{ where } i > 1$$