2020 ICPC Asia Taiwan Online Programming Contest

日期：109 年 10 月 07 日 星期三 18:00~21:00 教室：SEC506

NO.	學號	簽名	NO.	學號	簽名
	36.7 10711187	吳宇祥	36.2	10711106	邱文運
1	35.5 10711128	蘇聖博	36.4 6	10811226	陳冠秉
	36.1 10711137	[illegible]	35.8	10811120	[illegible]
	36.0 10711118	張淑盈	36.5	10711130	蘇聖益
2	34.9 10711105	何[illegible]	7 36.3	[illegible]	[illegible]
	35.9 10711131	吳俞誼	35.8	10711124	張照揚
35.5	10711117	李維	36.2	10711232	[illegible]
36.4 3	10711116	陳重甫	36.1 8	10711202	陳柏彰
36.0	10711206	林宣儀	36.3	10711246	簡瑋廷
36.4	10811228	林凱翔		10711238	陳郁嵐 36.2
36.2 4	[illegible]	[illegible]	9	10711245	吳易儒 36.3
36.3	10711117	黃[illegible]		10711204	許成誼 36.2
36.3	10611105	林建豪		10811131	吳丞恩 36
36.0 5	10611128	劉宏佑	10	10811237	李振偉 36.5
36.2	10611124	黃梓恩		10811117	薛映彤 36.0

2020 ICPC Asia Taiwan Online Programming Contest

日期：109 年 10 月 07 日 星期三 18:00~21:00 教室：SEC506

NO.	學號	簽名	NO.	學號	簽名
1	10811218	歐子婕 36.0	6		
	10811209	曾宥竹 35.7			
	10811206	林浚賢 35.8			
36.2 2	10711237	葉柘郡	7		
	10711270	潘沛晴 36.3			
	10711201	綽芸文 36.4			
35.9 3 36.0 36.2	10711241	陳智勻	8		
	10711231	楊筱禪			
	10711243	陳詩豪			
4			9		
5			10		

Problem A
Pac-Man

Time limit: 1 second

Memory limit: 1024 megabytes

Problem Description

Pac-Man is a maze-chase video game developed in 1980s. The player controls the character "Pac-Man" to eat dots in a maze while avoiding enemy characters "ghosts." All characters may move in four directions: up, down, left, right. The game ends when one of the following two conditions is met:

1. Pac-Man eats all dots in the maze. In this case, the player wins.
2. A ghost catches Pac-Man. In this case, the player loses.

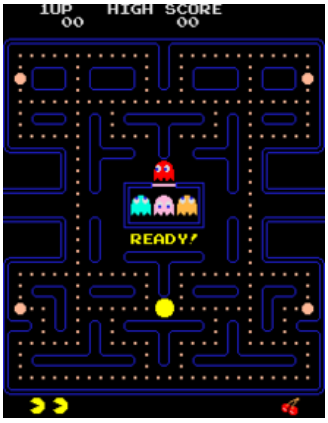


Figure 1: Pac-Man gameplay (image from Wikipedia)

Adam is learning how to create games with modern programming tools. To practice the skills, he tries to reproduce the Pac-Man game with some modification. In Adam's game, the playable character is a "ghost," and the enemy character is "Pac-Man." Since he changes the roles of the ghost and Pac-Man, he also changes the ending conditions of the game.

1. Pac-Man eats all dots in the maze. In this case, the player loses.
2. The ghost controlled by the player catches Pac-Man. In this case, the player wins.

Adam has almost developed the first full functioning version of his game. He wants to test his game and creates a simple stage for testing. The maze of the stage is based on a 10-by-10 grid. We label the cell at the intersection of row r and column c with (r, c). In this problem, rows and columns are numbered from 0 to 9. Each grid cell contains exact one dot. The exterior boundary of the grid are walls. No characters may move to the area outside of the grid. Inside the grid, there are no walls or obstacles. All characters may move freely from a cell to any cell adjacent to it. Note that two grid cells (r_1, c_1) and (r_2, c_2) are adjacent to each other if and only if $|r_1 - r_2| + |c_1 - c_2| = 1$.

Adam has to prepare the movements of Pac-Man for the testing. He needs a set of Pac-Man's trajectories with diversity, but any trajectory must satisfy the following requirements.

- Pac-Man eats all dots in the maze if it follows the trajectory.
- Pac-Man moves at most 10000 steps.

Adam needs your help to generate a trajectory starting at cell (x, y). Please write a program to generate a trajectory of Pac-Man satisfying all requirements above and starting at cell (x, y).

Input Format

The input has exactly one line which consists of two space-separated integers x and y. You are asked to generate a trajectory starting at cell (x, y).

Output Format

You must output a requested trajectory in the following format:

The trajectory is represented by $m + 1$ lines where m is the number of steps of the trajectory. The i-th line contains two integers r_i and c_i separated by exactly one space. The trajectory starts at the cell (r_1, c_1), and Pac-Man will be in cell (r_i, c_i) after moving $i - 1$ steps along the trajectory for $1 < i \leq m + 1$.

Technical Specification

- $m \leq 10000$
- $x, y, r_i, c_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ for all $i \in \{1, 2, \ldots, m + 1\}$.
- Cells (r_i, c_i) and (r_{i+1}, c_{i+1}) are adjacent to each other for all $i \in \{1, 2, \ldots, m\}$.
- $\{(r_1, c_1)\} \cup \{(r_2, c_2)\} \cup \cdots \cup \{(r_{m+1}, c_{m+1})\} = \{(r, c) : r \in \{0, 1, \ldots, 9\}, c \in \{0, 1, \ldots, 9\}\}$
- If there are multiple solutions, then you may output any of them.

Sample Input 1

```
0 0
```

Sample Output 1

```
0 0
0 1
0 2
...
...
...
9 2
9 1
9 0
```

Note

The sample output section does not contain the correct output, since it ignores a large part of the answer. Please download the correct sample test cases from the judge system.

Problem B
Folding

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

There is a transparent tape. Its length is exact one meter (10^9 nanometers). In this problem, all numbers are integers, and we use a number to denote a position on the tape. The number p denote the position of the point has a distance p nanometers from the head of the tape.

Bob is a master dyer, so he can color the tape precisely in nanometer scale. He colors two sectors $[p_1, q_1]$ and $[p_2, q_2]$ into red. The color of the tape within the range between p_1 and q_1 is red. The color of the tape within the range between p_2 and q_2 is also red. And the rest parts of the tape remain transparent.

To verify Bob's skill, we ask Ben, the tape folding master, to help us. Ben can fold the tape perfectly at any position. If Ben fold the tape at x, then the new position of a certain point p will be one of the following cases.

- If $p = x$, then it becomes the new head of the tape, i.e, it becomes 0.
- If $p > x$, then it becomes $p - x$.
- If $p < x$, then it becomes $x - p$.

After Ben folds the tape, we measure the total length of the red part of the new tape. If the red part has the expected length, then we will believe Bob and Ben are both masters in their skills. Obviously, the color of some position of the new tape is determined by the colors of the corresponding positions of the old tape. A position of the new tape is colored in red if one of the corresponding positions in the old tape is colored in red.

Bob has already colored the tape, and Ben has proposed the positions to be folded. Please write a program to compute the expected lengths colored in red.

Input Format

The first line contains four space-separated integers p_1, q_1, p_2 and q_2. Bob has colored the sectors $[p_1, q_1]$ and $[p_2, q_2]$. The second line contains an integer q indicating that Ben has made q proposals. Each of the remaining q lines contains an integer x indicating the positions to be folded by Ben. Note that the q proposals are independent to each other. There is only one folding point in one proposal.

Output Format

For each position, output the expected total length of the new tape that are colored in red.

Technical Specification

- $0 \leq p_1 < q_1 < p_2 < q_2 \leq 10^9$
- $0 \leq x \leq 10^9$
- $q \leq 10^6$

Sample Input 1

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

Sample Output 1

```
3
2
3
3
2
3
3
3
3
3
```

Problem C
Circles

Time limit: 8 seconds

Memory limit: 1024 megabytes

Problem Description

There are n magical circles on a plane. They are centered at $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, respectively. In the beginning, the radius of each circle is 0, and the radii of all magical circles will grow at the same rate. When a magical circle touches another, then it stops growing. Write a program to calculate the total area of all magical circles at the end of growing.

Input Format

The first line contains an integer n to indicate the number of magical circles. The i-th of the following n lines contains two space-separated integers x_i and y_i indicating that the i-th magical circle is centered at (x_i, y_i).

Output Format

Output the total area of the circles.

Technical Specification

- $2 \leq n \leq 2000$
- $x_i, y_i \in [-10^9, 10^9]$ for $i \in \{1, 2, \ldots, n\}$.
- All (x_i, y_i)'s are disinct points.
- A relative error of 10^{-6} is acceptable.

Sample Input 1

```
4
0 0
1 0
1 1
0 1
```

Sample Output 1

```
3.14159265359
```

Sample Input 2

```
3
0 0
0 1
2 0
```

Sample Output 2

```
8.639379797371932
```

Almost blank page

Problem D
Last Will

Time limit: 1 second

Memory limit: 1024 megabytes

Problem Description

David is a farmer and has a large farm. The shape of the farm is a square. A sqaure is a quadrilaterial that has four equal sides and four equal angles. The length of any side of David's farm is one kilometer, so the area of his farm is slightly greater than the total area of 140 standard football fields.

David is old and very ill. He feels that his days are numbered. Worrying that his spouse Dora and their three children, Alice, Bob, and Cliff, will have a dispute over the ownership of the farm after he passes away, he plans to divide the farm into four parts, and then to allocate each part to one of his family members. He decides to write his last will as follows.

1. Assume that the shape of the farm is a sqaure $ABCD$ where $A = (0, 0)$, $B = (1, 0)$, $C = (1, 1)$, $D = (0, 1)$.
2. Let $E = (0.5, 0)$, $F = (1, 0.5)$, $G = (0.5, 1)$, $H = (0, 0.5)$ be the midpoints of $\overline{AB}$, $\overline{BC}$, $\overline{CD}$, $\overline{DA}$, respectively.
3. Let $area(PRQS)$ to denote the area of the quadrilaterial $PQRS$.
4. Please find a point X *strictly inside* the square $ABCD$ such that
$$area(AEXH) : area(BFXE) : area(CGXF) = p : q : r$$
Note that X cannot be on the boundary of the square $ABCD$.
5. Allocate the land in $AEXH$, $BFXE$, $CGXF$, $DHXG$ to Alice, Bob, Cliff and Dora, respectively.

David is still adjusting the numbers p, q, r, and his lawyer, Reed, has to read David's last will carefully. Reed finds that it is impossible to find such point X if David gives an improper set of the numbers such as $p = 1, q = 2, r = 1$. However, there are proper sets of the numbers p, q, r that allow us to find the point X satisfying David's last will. For instance, let $p : q : r = 2 : 3 : 2$, the following figure shows a possible position of X.

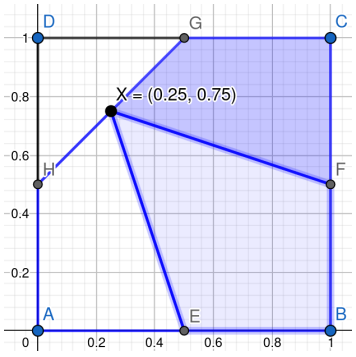


Figure 2: $area(AEXH) : area(BFXE) : area(CGXF) = 2 : 3 : 2$

Please write a program to help Reed to determine whether it is possible to find a point X satisfying David's last will for a given set of numbers p, q, r. If possible, please output one possible position of X to Reed.

Input Format

The input contains one line only. The line contains three space-separated positive integers p, q, r.

Output Format

If there does not exist a point X satisfying David's last will, then output `-1`. Otherwise, output two irreducible fractions x and y such that (x, y) can be the point X satisfying David's last will. You must output an irreducible fraction $t = \frac{n}{d}$ as n/d with a positive denominator and use exactly one space to separate x and y.

Note: the numerator and the denominator of any irreducible fraction are integers and do not have common divisors other than 1 and -1.

Technical Specification

- $p, q, r \in \{1, 2, \ldots, 10^6\}$

Sample Input 1

```
1 1 1
```

Sample Output 1

```
1/2 1/2
```

Sample Input 2

```
1 2 1
```

Sample Output 2

```
-1
```

Sample Input 3

```
2 3 2
```

Sample Output 3

```
1/4 3/4
```

Problem E
Eric's Work

Time limit: 5 seconds
Memory limit: 1024 megabytes

Problem Description

A binary string is a string consisting of only 0's and 1's. Elsa, Eric's boss, gave him a binary string s of length 20 and asked him to modify s into another binary string t within D days.

Eric really hates this task and therefore never modifies more than one character in a day. However, being forced to show Elsa the daily progress, Eric must modify some characters of the string every day. That means, the only possible way for Eric is to modify exact one character in each day before he finishes the task.

It is obviously cheating to have a character changed to something other than 0 and 1. Moreover, Eric will be caught cheating if the string is modified into the same binary string twice since Elsa has a good memory. That is, before the string is modified into t, all modifications result in unique strings. Note that Eric cannot modify the string into s which is the string given by Elsa, either.

Eric wants to spend as much time as possible. He is wondering if he can spend exact D days to have the string s modified into t. Please write a program to help Eric.

Input Format

The input contains three lines. The first line contains a binary string s. The second line contains a binary string t. The third line contains an integer D. Elsa asked Eric to modify the binary string s into t within D days.

Output Format

If there is no way to achieve what Eric wants, output `-1`. Otherwise, output D lines to represent one possible way. The i-th line contains a binary string, the result of the modification on the i-th day.

Technical Specification

- The strings s and t consist of only 0's and 1's.
- The length of s and the length of t are both 20.
- $1 \leq D \leq 500000$
- If there are multiple solutions, then you may output any of them.

Sample Input 1

```
00000000000000000000
11111111111111111111
5
```

Sample Output 1

```
-1
```

Sample Input 2

```
00000000001111111111
10000000001111111111
5
```

Sample Output 2

```
00000000001111111110
00000001001111111110
10000001001111111110
10000000001111111110
10000000001111111111
```

Problem F
Homework

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

There are n children (numbered from 1 to n) learning the arithmetic operations, which include *addition* "+", *subtraction* "−", *multiplication* "×", and *division* "÷" on rational numbers.

In the beginning, each child has a paper sheet with only a zero on it. Their teacher, Frank, will then give them q operations. The i-th operation consists of an operator c_i and an integer x_i. The children numbered $\ell_i, \ell_i + 1, \ldots, r_i$ have to append the operator c_i and the integer x_i to their paper sheets. After that, every child has an expression on their sheet to be evaluated.

For example, suppose that $n = 3$, $q = 2$, c_1 is "+", $x_1 = 1$, $\ell_1 = 1$, $r_1 = 2$, c_2 is "−", $x_2 = 2$, $\ell_2 = 2$, and $r_2 = 3$. The expressions on the sheets are are $0+1$, $0+1-2$ and $0-2$ for children 1, 2 and 3, respectively.

Since Frank is really lazy and wants to verify the answers quickly, he asks you to calculate the sums of the values of all children's expressions. Suppose that the value of the expression assigned to child i is $\frac{a_i}{b_i}$, then the value will be $a \times b^{-1} \bmod 10^9 + 7$ instead, where b^{-1} denotes the integer satisfying $b \times b^{-1} \equiv 1 \pmod{10^9 + 7}$. If the sum is not in $[0, 10^9 + 7)$, then the sum modulo $10^9 + 7$ should be returned to Frank.

Note: The arithmetic operations has PEMDAS rule, that is, multiplications and divisions should be evaluated before evaluating additions and subtraction.

Input Format

The first line consists of two space-separated integers n and q. The i-th of the following q lines consists of four space-separated tokens ℓ_i, r_i, c_i, x_i. For the sake of convenience, `*` and `/` are used to represent the multiplication and the division operators, respectively.

Output Format

Output the number that you should return to Frank.

Technical Specification

- $1 \le n \le 10^5$
- $1 \le q \le 10^5$
- $\ell_i, r_i \in [1, n]$ for all $1 \le i \le q$.
- $c_i \in \{$`+`, `-`, `*`, `/`$\}$ for all $1 \le i \le q$.
- For all $1 \le i \le q$, $x_i = 0$ implies that c_i is not `/`.
- $x_i \in [0, 10^9 + 7)$ for all $1 \le i \le q$.

Sample Input 1

```
3 2
1 2 + 1
2 3 - 2
```

Sample Output 1

```
1000000005
```

Problem G
Garden

Time limit: 10 seconds
Memory limit: 1024 megabytes

Problem Description

There is a rectangle garden in front of Gina's house. The garden can be seen as an n-by-m rectangular grid. All grid cells are identical squares, and two grid cells are considered adjacent if they share an common edge.

Gina loves cacti and wants to plant as many cacti as possible in the garden. However, there are some constraints on planting cacti.

- The soil can be too wet in some of the cells and therefore is not suitable for cacti. Gina cannot plant cacti in those cells.
- Since the soil in each cell is not fertile enough to grow two or more cacti, Gina may plant at most one cactus in a cell.
- At most one cactus can be planted in any pair of adjacent cells. Otherwise, the cacti in those cells may be harmed by their neighbor's thorns.

Please write a program to help Gina calculate the maximum possible number of cacti can be planted, and also a way of plantings that meet the listed constraints.

Input Format

The first line contains two space-separated integers n and m meaning the garden is an n-by-m grid. Then, each of the following n lines contains a string of m characters. These characters are either '`.`' or '`*`'. The j-th character of the i-th of these lines indicates whether the soil in the grid cell on the i-th row and the j-th column is suitable for planting a cactus. '`.`' means it is suitable, and '`*`' means it is not suitable.

Output Format

First, output the maximum possible number of cacti on the first line. Then, output n lines, each line containing a string of m characters. Each of the characters must be one of '`.`', '`*`' and '`C`'. The j-th character of the i-th of these lines indicates the status of the grid cell on the i-th row and the j-th column. A '`C`' means a cactus should be planted in that particular cell, and the other cells should be identical to the corresponding position of the input.

Technical Specification

- $1 \leq nm \leq 10^5$
- If there are more than one possible way of planting, any of them will be accepted.

Sample Input 1

```
3 3
*.*
...
*.*
```

Sample Output 1

```
4
*C*
C.C
*C*
```

Sample Input 2

```
2 4
*..*
....
```

Sample Output 2

```
3
*C.*
C.C.
```

Problem H
In The Name Of Confusion

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

> There's no such thing as public opinion.
>
> *Jordan Ellenberg, American Mathematician*

In K City lives n residents who want to build a connection network with each other. However, some residents want the network wires colored black while the others want the wires colored white. The opinion of resident i can be quantified as a number a_i. If we build a network wire between residents i and j, the cost of this wire will be $a_i \times a_j$.

The mayor of K City wants to build a network such that:

1. There are exactly $n - 1$ wires used.
2. For any two different residents i and j, there exists a sequence $p_1, \cdots, p_k$ such that $p_1 = i$, $p_k = j$ and residents p_ℓ and $p_{\ell+1}$ share a wire for $1 \leq \ell < k$.

In other words, the network should be a tree.

You, the renowned mathematician of K City, want to know not only the *minimum* cost to build the network. In the name of confusion, you also want to know the *maximum* cost!

Input Format

The first line begins with a number n indicating the number of residents. The second line contains n numbers $a_1, a_2, \ldots, a_n$. The opinion of resident i is the quantified as a_i.

Output Format

Output two numbers separated by a blank in a line. The numbers are the *minimum* cost and the *maximum* cost to build the network, respectively. Since the absolute value of the costs may be extremely large, you have to modulo the answer with $10^9 + 7$. Please note that the modulo of a number (defined by Donald Knuth) is $a \bmod b = a - b\lfloor \frac{a}{b} \rfloor$. The output number should be non-negetive.

Technical Specification

- $1 \leq n \leq 10^6$
- $|a_i| \leq 10^6$

Sample Input 1

```
10
-5 -10 -7 -7 -3 -1 -7 -5 -8 -6
```

Sample Output 1

```
58 490
```

Sample Input 2

```
10
-5 1 2 -2 -1 1 -5 5 -10 6
```

Sample Output 2

```
999999779 183
```

Sample Input 3

```
10
0 0 0 0 0 0 0 0 0 0
```

Sample Output 3

```
0 0
```

Sample Input 4

```
10
10 8 9 3 8 8 0 5 3 10
```

Sample Output 4

```
0 540
```

Problem I
Site Score

Time limit: 1 second
Memory limit: 1024 megabytes

Problem Description

Teams from variaous universities compete in ICPC regional contests for tickets to the ICPC World Finals. The number of tickets allocated to every regional contest may be different. The allocation method in our super region, Asia Pacific, is based on a parameter called site score.

Site scores will only count teams and universities solving at least one problem, in the regional contest or its preliminary contest TOPC. In 2020, the formula for calculating the site score of the Taipei-Hsinchu regional contest is much simpler than past years. Let

- U_R be the number of universities solving at least one problem in the regional contest.
- T_R be the number of teams solving at least one problem in the regional contest.
- U_O be the number of universities solving at least one problem in TOPC.
- T_O be the number of teams solving at least one problem in TOPC.

The site score of 2020 Taipei-Hsinchu regional contest will be $56U_R+24T_R+14U_O+6T_O$. Please write a program to compute the site score of the 2020 Taipei-Hsinchu regional contest.

Input Format

The input has only one line containing four blank-separated positive integers U_R, T_R, U_O, and T_O.

Output Format

Output the site score of the 2020 Taipei-Hsinchu regional contest.

Technical Specification

- $0 < U_R \leq T_R \leq 120$
- $0 < U_O \leq T_O \leq 1000$

Sample Input 1

```
1 1 1 1
```

Sample Output 1

```
100
```

Sample Input 2

```
1 10 100 1000
```

Sample Output 2

```
7696
```

Note

The problem statement is fiction. The real site score has a different formula.

Almost blank page

Problem J
Table Tennis

Time limit: 5 seconds
Memory limit: 1024 megabytes

Problem Description

Alex is attending the first edition of Robotic World Championship of Table Tennis. A competition that have all of the matches having the same rules listed below:

- A match shall consist of the best of 7 games, i.e., the results of matches must be 4 games to k, where $0 \leq k \leq 3$.
- A game shall be won by the player first scoring 11 points unless both players score 10 points, when the game shall be won by the first player subsequently gaining a lead of 2 points. For example, a game can be won at scores like 11-5, 11-9 or 12-10, but not 10-5 or 11-10.
- After each 2 points have been scored the receiving player shall become the serving player and so on until the end of the game, unless both players score 10 points, when the sequences of serving and receiving shall be the same but each player shall serve for only 1 point in turn. That is, the servicing order of the first 20 points is `AABBAABBAABBAABBAABB`, and will be followed by `ABABAB`... if necessary.
- The player serving first in a game shall receive first in the next game of the match.

Experience tells that when two robots clashes into each other, the variances affecting their winning chances can be simplified to who's serving for the point. This is due to the performances of the robots are physically consistent and won't be affected mentally.

Alex have listed some of the possible matchups, simplified to the winning chance of each servicing point of the robots, for you. Now it is your job to help him calculate the winning chance of each match for them.

Input Format

The first line of the input consists of a single number T, indicating that there will be T test cases following.

Each of the following test case consists of two space-separated real numbers P_A and P_B in one line, where P_A denotes the Robot A's chance of winning the point when A is serving and P_B denotes the Robot B's chance of winning the point when B is serving.

The Robot A always serves first in the very first game of the match.

Output Format

For each test case, output one real number in one line: the winning chance of A.

Technical Specification

- $T \leq 100$
- $0 \leq P_A \leq 1$ and has at most 2 digits after the decimal point in the input.
- $0 \leq P_B \leq 1$ and has at most 2 digits after the decimal point in the input.
- $0 < P_A + P_B < 2$
- The answer will be considered correct if it is within an absolute error of 10^{-8} of the correct answer.

Sample Input 1

```
3
1 0
0.5 0.5
0.00 1.00
```

Sample Output 1

```
1
0.5
0.000000000
```

References

The rules are revised from the Chapter 2 "The Laws of Table Tennis" of The International Table Tennis Federation (ITTF) Handbook 2020.

2020 ICPC Asia Taiwan Online Programming Contest preliminary results - not final

Filter

RANK	TEAM	SCORE	A	B	C	D	E	F	G	H	I	J
1	2020 TOPC **BBQube**	**10** 947	20 1 try	34 1 try	17 1 try	148 2 tries	174 1 try	107 2 tries	156 2 tries	81 1 try	4 1 try	146 1 try
2	**waynedisonitau123**	**9** 601	23 2 tries	26 1 try	34 2 tries	32 1 try	 2 tries	177 3 tries	69 2 tries	63 1 try	2 1 try	75 1 try
3	**CRyptoGRapheR**	**9** 786	15 1 try	79 2 tries	68 1 try	91 1 try	171 1 try	 	146 2 tries	115 1 try	4 1 try	57 1 try
4	**kiseki**	**8** 494	8 1 try	27 1 try	38 1 try	54 1 try	 	 4 tries	146 2 tries	75 1 try	9 1 try	97 2 tries
5	**28**	**8** 708	8 1 try	79 2 tries	40 1 try	136 2 tries	 	174 2 tries	 	34 1 try	2 1 try	155 2 tries
6	**NoName**	**8** 747	7 1 try	72 2 tries	64 2 tries	71 2 tries	 	 	135 2 tries	77 2 tries	3 1 try	178 3 tries
7	**BAN**	**8** 783	12 1 try	61 2 tries	56 1 try	87 3 tries	 	 	150 5 tries	71 1 try	4 1 try	162 3 tries
8	**NCTU_a**	**6** 292	10 1 try	18 1 try	96 1 try	51 1 try	 	 	 	114 1 try	3 1 try	 1 try
9	**NCTU_Eclipse**	**6** 360	17 1 try	41 1 try	78 1 try	138 2 tries	 	 	 2 tries	60 1 try	6 1 try	
10	**UTAhahaha**	**6** 390	22 1 try	64 1 try	99 5 tries	74 1 try	 	 2 tries	 	41 1 try	10 1 try	 1 try
11	**LYB3**	**6** 435	21 1 try	133 2 tries	47 1 try	114 1 try	 	 	 	89 1 try	11 1 try	 1 try
12	**NCTU_Capooooooo**	**6** 436	17 1 try	35 1 try	101 4 tries	62 1 try	 	 	 1 try	138 2 tries	3 1 try	 3 tries
13	**OwO**	**6** 452	21 1 try	119 2 tries	89 2 tries	98 1 try	 	 	 	72 1 try	13 1 try	
14	**110NTNU**	**6** 482	9 1 try	81 2 tries	113 3 tries	87 1 try	 	 	 	130 1 try	2 1 try	
15	**ABCodeboook**	**6** 520	44 2 tries	50 1 try	68 2 tries	 	 	 	 	99 1 try	5 1 try	174 3 tries
16	**NCTU_Banana**	**5** 262	26 1 try	68 1 try	 7 tries	75 2 tries	 	 	 3 tries	50 2 tries	3 1 try	
17	**NCTU__**	**5** 310	21 1 try	47 1 try	 	143 1 try	 	 	 	95 1 try	4 1 try	
18	**EX-panda**	**5** 338	55 2 tries	27 1 try	 5 tries	95 1 try	 	 	 	137 1 try	4 1 try	
19	**NCU**	**5** 372	17 1 try	55 2 tries	 	152 1 try	 	 	 	123 1 try	5 1 try	
20	**HongLongLongLong**	**5** 387	25 1 try	49 2 tries	 1 try	136 2 tries	 	 	 	133 1 try	4 1 try	
21	**zappers**	**5** 401	35 1 try	81 1 try	122 1 try	 1 try	 1 try	 	 1 try	157 1 try	6 1 try	
22	**NTNU_ch1W4w4**	**5** 429	33 2 tries	64 2 tries	 3 tries	159 2 tries	 	 	 	106 1 try	7 1 try	
23	**C++S**	**5** 440	39 1 try	74 1 try	134 3 tries	 	 	 	 	114 2 tries	19 1 try	 4 tries
24	**BITreap**	**5** 533	16 1 try	163 3 tries	140 1 try	149 2 tries	 	 	 	 	5 1 try	
25	**NCTU_HongLongLong**	**5** 555	87 1 try	126 3 tries	 3 tries	59 3 tries	 	 	 	179 2 tries	4 1 try	
26	**ToBeContinued**	**4** 146	9 1 try	36 2 tries	 1 try	67 1 try	 	 	 	 	14 1 try	 1 try
27	**bz\pileafd**	**4** 197	29 2 tries	51 1 try	 3 tries	91 1 try	 	 	 	 4 tries	6 1 try	 3 tries

RANK		TEAM	SCORE		A	B	C	D	E	F	G	H	I	J
28	♡	**NCTU_3.12B**	**4**	215	21 1 try	61 2 tries	2 tries	86 2 tries				3 tries	7 1 try	
29	♡	**NCTU_White**	**4**	252	29 1 try	120 1 try	1 try				1 try	97 1 try	6 1 try	
30	♡	**NTUT_Kn1ghts**	**4**	262	56 1 try	135 1 try			66 1 try		2 tries		5 1 try	
31	♡	**NCTU_LoTaTea**	**4**	274	50 3 tries	62 1 try				2 tries		116 1 try	6 1 try	
32	♡	**Rabbit Foot**	**4**	275	20 1 try	63 1 try		161 2 tries			2 tries	4 tries	11 1 try	
33	♡	**CCU_hahahaha**	**4**	296	23 1 try	65 1 try	2 tries	119 5 tries				4 tries	9 1 try	
34	♡	**RedThunder**	**4**	308	37 2 tries	132 3 tries		75 1 try				2 tries	4 1 try	1 try
35	♡	**NTPU_kite**	**4**	310	34 1 try	55 1 try		115 6 tries			1 try		6 1 try	
36	♡	**NCTU_Daisangen**	**4**	313	38 2 tries	142 4 tries		44 1 try					9 1 try	
37	♡	**NCTU_KokushiMusou**	**4**	315	33 2 tries	120 4 tries		57 2 tries				1 try	5 1 try	
38	♡	**NCU_309**	**4**	331	46 1 try	135 3 tries		103 1 try					7 1 try	
39	♡	**Really Simply Accepted**	**4**	531	25 1 try	178 3 tries		3 tries			1 try	162 7 tries	6 1 try	
40	♡	**NCTU_Pusheen**	**3**	100	38 1 try	37 2 tries	3 tries	2 tries				6 tries	5 1 try	
41	♡	**the gourmet of the desert**	**3**	112	17 1 try	89 1 try		10 tries					6 1 try	
42	♡	**SAI_sixtieth**	**3**	127	18 1 try	4 tries	4 tries	64 2 tries					5 2 tries	
43	♡	**MCU_Shark**	**3**	142	18 1 try	97 2 tries	1 try					3 tries	7 1 try	
44	♡	**GWAWA**	**3**	145	38 1 try	82 2 tries							5 1 try	
45	♡	**Anti-Accept**	**3**	148	33 1 try	86 2 tries		2 tries	1 try			2 tries	9 1 try	
46	♡	**NTNU_WWW**	**3**	181	62 1 try	105 1 try						2 tries	14 1 try	
47	♡	**CDJ**	**3**	184	54 1 try	96 2 tries							14 1 try	
48	♡	**SHAAAAARK**	**3**	206	111 2 tries	48 2 tries		2 tries					7 1 try	
49	♡	**FJCU_Return0**	**3**	210	66 1 try	1 try		139 1 try				1 try	5 1 try	
50	♡	**CSDD**	**3**	215	19 1 try	148 3 tries	1 try						8 1 try	
51	♡	**NSYSU_Ant**	**3**	219	103 1 try			45 1 try	1 try			1 try	71 1 try	
52	♡	**NTNU_import_magic**	**3**	222	31 1 try	145 3 tries						2 tries	6 1 try	
53	♡	**WeWantWin**	**3**	260	117 2 tries	2 tries	1 try	1 try				108 1 try	15 1 try	
54	♡	**HmmmHuhh**	**3**	263	42 3 tries	131 3 tries		1 try					10 1 try	
55	♡	**CCU_TOOL_MAN**	**3**	279	89 2 tries	159 1 try		2 tries					11 1 try	
56	♡	**TTU02**	**3**	387	125 4 tries	149 3 tries							13 1 try	
57	♡	**NUK_4+1BadBad**	**2**	23	17 1 try			1 try					6 1 try	
58	♡	**BearHaoEn**	**2**	24	19 1 try	1 try		2 tries					5 1 try	

RANK		TEAM	SCORE		A	B	C	D	E	F	G	H	I	J
59	♡	**TCUMI Need Money**	**2**	29	22 1 try	 1 try							7 1 try	
60	♡	**NTCU_NONAMETEAM**	**2**	31	26 1 try	 4 tries			 1 try				5 1 try	
61	♡	**Acpc**	**2**	35	31 1 try	 1 try						 3 tries	4 1 try	
62	♡	**SendTreePay**	**2**	42	35 1 try	 2 tries							7 1 try	
63	♡	**NTPU_boomer**	**2**	44	37 1 try	 3 tries	 1 try					 1 try	7 1 try	
64	♡	**NUK_EhGanPangolin**	**2**	54	47 1 try	 6 tries	 2 tries						7 1 try	
65	♡	**XDDD**	**2**	56	40 1 try						 2 tries		16 1 try	
66	♡	**Three_dogs**	**2**	68	41 2 tries	 4 tries		 1 try					7 1 try	
67	♡	**MCU_Unicorn**	**2**	69	43 2 tries	 9 tries							6 1 try	
68	♡	**NCTU_Chinroutou**	**2**	73	44 2 tries	 4 tries			 2 tries				9 1 try	
69	♡	**programming fly high high**	**2**	73	52 1 try	 2 tries							21 1 try	
70	♡	**For The Point**	**2**	75	63 1 try	 3 tries					 4 tries		12 1 try	
71	♡	**NSYSU_Dog**	**2**	79	72 1 try	 1 try					 3 tries		7 1 try	
72	♡	**CCU_Snorlax**	**2**	80	56 2 tries	 3 tries		 3 tries					4 1 try	 4 tries
73	♡	**bits/stdc++.h**	**2**	81	52 1 try	 3 tries							29 1 try	
74	♡	**Asterisk**	**2**	82	53 2 tries	 4 tries					 2 tries		9 1 try	
75	♡	**Senpai Sayounara**	**2**	89	57 2 tries	 1 try	 1 try	 2 tries	 1 try			 1 try	12 1 try	
76	♡	**123**	**2**	91	83 1 try						 3 tries		8 1 try	
77	♡	**no idea**	**2**	104	89 1 try	 3 tries							15 1 try	
78	♡	**whatever**	**2**	104	96 1 try	 2 tries							8 1 try	
79	♡	**c2hBcms=**	**2**	109	81 2 tries						 2 tries		8 1 try	
80	♡	**NCU_Coder**	**2**	116	 3 tries	69 3 tries			 2 tries			 1 try	7 1 try	
81	♡	**FJCU-ITTE**	**2**	128	121 1 try						 2 tries		7 1 try	
82	♡	**300kgTaiwanPig**	**2**	142	121 1 try				 2 tries				21 1 try	
83	♡	**Mef**	**2**	144	92 3 tries								12 1 try	
84	♡	**Mine englishes bed**	**2**	152	128 2 tries			 1 try					4 1 try	
85	♡	**CCU_Melody**	**2**	154	105 3 tries	 1 try	 1 try	 1 try	 1 try				9 1 try	
86	♡	**maple_drizzle**	**2**	162	107 3 tries								15 1 try	
87	♡	**Team Name**	**2**	162	124 2 tries	 4 tries							18 1 try	
88	♡	**MaseLabYONG 2**	**2**	165	98 2 tries								47 1 try	
89	♡	**CHU_MapoTofu**	**2**	167	39 1 try	 3 tries			 2 tries				128 1 try	

RANK		TEAM	SCORE		A	B	C	D	E	F	G	H	I	J
90	♡	**TeamACE**	**2**	170	116 3 tries		2 tries						14 1 try	
91	♡	**NSYSU_Bat**	**2**	186	162 1 try								24 1 try	
92	♡	**YesOrNo**	**2**	206	151 2 tries	3 tries			1 try				15 2 tries	2 tries
93	♡	**AKBandai**	**2**	215	172 3 tries	3 tries		1 try					3 1 try	
94	♡	**MCU_Alpin**	**2**	239	130 6 tries								9 1 try	
95	♡	**AmongUs**	**2**	248	177 3 tries								31 1 try	
96	♡	**Linzoma**	**2**	258	179 4 tries	4 tries					1 try		19 1 try	
97	♡	**IndexO**	**2**	285	179 6 tries	4 tries							6 1 try	
98	♡	**GAMANNEI**	**1**	4	3 tries								4 1 try	
99	♡	**WA_only**	**1**	5	6 tries								5 1 try	
100	♡	**CCU_OreO**	**1**	6	4 tries	1 try							6 1 try	
101	♡	**345**	**1**	7		5 tries					1 try		7 1 try	
	♡	**BRM**	**1**	7	2 tries								7 1 try	1 try
	♡	**CCU_Paofu**	**1**	7	4 tries		2 tries		2 tries				7 1 try	
104	♡	**MCU_Delta**	**1**	8				3 tries					8 1 try	
	♡	**MCU_Hydra**	**1**	8	1 try				1 try				8 1 try	
106	♡	**CJCU_AIRC**	**1**	9									9 1 try	
	♡	**memetour**	**1**	9	5 tries	1 try							9 1 try	
108	♡	**Hello world**	**1**	10		2 tries			1 try		2 tries		10 1 try	
109	♡	**floccinaucinihilipilifiCATion**	**1**	11	1 try								11 1 try	
	♡	**JZHW**	**1**	11				1 try			1 try		11 1 try	
	♡	**Moomin**	**1**	11	3 tries								11 1 try	
	♡	**NUK_JellyLegion**	**1**	11		1 try	1 try						11 1 try	
	♡	**Salt fish**	**1**	11	7 tries								11 1 try	
114	♡	**CCU Nocturnal penile tumescencea**	**1**	12			1 try	3 tries			1 try		12 1 try	
	♡	**MCU_AC**	**1**	12	2 tries								12 1 try	
	♡	**MCU_Aqua**	**1**	12	2 tries		1 try						12 1 try	
	♡	**NUK_10755**	**1**	12	2 tries				3 tries				12 1 try	
118	♡	**612-1**	**1**	13	2 tries	3 tries							13 1 try	
	♡	**FJCU_C**	**1**	13							3 tries		13 1 try	
	♡	**FJU GGWP/FF**	**1**	13									13 1 try	

RANK		TEAM	SCORE		A	B	C	D	E	F	G	H	I	J
	♡	**New_Game**	**1**	13	5 tries		2 tries	2 tries					13 1 try	
	♡	**NSYSU_PLAY3**	**1**	13	1 try								13 1 try	
	♡	**SlackOff**	**1**	13	5 tries								13 1 try	
124	♡	**HCCISGOOD**	**1**	14	2 tries								14 1 try	
	♡	**iNEMU**	**1**	14					1 try		2 tries	4 tries	14 1 try	
	♡	**MCU_Tesla**	**1**	14	1 try								14 1 try	
127	♡	**\JIN-SE-KUANG-FENG/**	**1**	15	1 try						9 tries		15 1 try	
128	♡	**FuJenAddPoint**	**1**	16	5 tries								16 1 try	2 tries
	♡	**It iS Fun !!!**	**1**	16		3 tries		1 try					16 1 try	
130	♡	**Spamail**	**1**	18	4 tries								18 1 try	
131	♡	**PU_shuiguojupanwushiG**	**1**	19	1 try		2 tries						19 1 try	1 try
132	♡	**c.cpp**	**1**	20									20 1 try	
133	♡	**steel**	**1**	21									21 1 try	
134	♡	**Natsuyasumi no owari**	**1**	23	1 try								23 1 try	
135	♡	**NSYSU_Cat**	**1**	24	3 tries	1 try							24 1 try	
136	♡	**Sally's Lab**	**1**	25	1 try								25 1 try	
137	♡	**UCCU**	**1**	27	1 try	11 tries						4 tries	27 1 try	
138	♡	**ABCDEFG**	**1**	30	1 try								30 1 try	
139	♡	**DforDing**	**1**	33		5 tries							33 1 try	
140	♡	**Comprehension**	**1**	34									34 1 try	
141	♡	**MaseLabYONG**	**1**	37									37 1 try	
142	♡	**JANuary**	**1**	44	7 tries								44 1 try	
143	♡	**612-2**	**1**	124		4 tries							124 1 try	
144	♡	**HTL**	**1**	147					3 tries				147 1 try	
145	♡	**STOS**	**1**	167									87 5 tries	
146	♡	**NSYSU_potplant**	**0**	0										
		SUMMARY	**395**		96 140 0 7min	53 167 0 18min	18 69 0 17min	35 71 0 32min	3 28 0 66min	3 12 0 107min	6 61 0 69min	28 60 0 34min	145 6 0 2min	8 33 0 57min
1	♡	**NTUB-001**	**0**	0										
	♡	**NTUB-002**	**0**	0										
		SUMMARY	**0**		0 0 0 n/a	0 0 0 n/a	0 0 0 n/a	0 0 0 n/a	0 0 0 n/a	0 0 0 n/a	0 0 0 n/a	0 0 0 n/a	0 0 0 n/a	0 0 0 n/a

Categories	Cell colours
2020 TOPC	Solved first
Observers	Solved
	Tried, incorrect
	Tried, pending
	Untried

Last Update: Wed 07 Oct 2020 21:37:12 CST
using DOMjudge

2020 ICPC Asia Taipei-Hsinchu Regional Contest preliminary results - not final

RANK	TEAM	SCORE	PA	PB	PC	PD	PE	PF	PG	PH	PI	PJ	PK	PL	PM
1	**waynedisonitau123** National Taiwan University	**10** 1214	2 1 try	14 1 try	221 1 try	 6 tries	179 3 tries	126 1 try	189 4 tries	34 1 try	24 1 try		200 7 tries		5 1 try
2	**CRyptoGRapheR** National Taiwan University	**10** 1309	4 1 try	17 1 try	239 1 try	 1 try	222 1 try	129 1 try	275 4 tries	57 1 try	36 1 try		192 2 tries		38 2 tries
3	**kiseki** National Taiwan University	**10** 1519	4 1 try	82 2 tries	102 1 try		161 1 try	94 1 try	266 4 tries	49 1 try	257 4 tries		284 4 tries		20 1 try
4	**UTAhahaha** National Taiwan University	**9** 1017	3 1 try	12 1 try		 3 tries	134 1 try	132 4 tries	264 1 try	30 1 try	106 2 tries		180 4 tries		16 1 try
5	**BBQube** National Taiwan University	**9** 1033	6 1 try	31 1 try			86 2 tries	117 1 try	237 4 tries	55 1 try	77 1 try		272 4 tries		12 1 try
6	**ABCodeboook** National Taiwan University	**8** 727	5 1 try	60 1 try	190 1 try		126 2 tries	150 1 try	 5 tries	44 1 try	97 2 tries				15 1 try
7	**NCTU_a** National Chiao Tung University	**7** 539	4 1 try	12 1 try			145 3 tries	121 1 try	 3 tries	91 1 try	71 3 tries				15 1 try
8	**NCTU_Capooooooo** National Chiao Tung University	**7** 671	3 1 try	28 1 try	185 2 tries		 6 tries	211 3 tries		56 1 try	84 2 tries				24 1 try
9	**LYB3** National Taiwan University	**7** 683	7 1 try	33 1 try		 9 tries	264 3 tries	136 1 try		60 2 tries	107 1 try				16 1 try
10	**EX-panda** National Tsing Hua University	**7** 734	4 1 try	59 1 try	168 1 try		145 1 try	196 1 try		140 1 try					22 1 try
11	**BAN** National Taiwan University	**6** 396	2 1 try	88 1 try	 1 try		65 1 try	131 1 try	 2 tries	95 1 try	 9 tries		 1 try		15 1 try
12	**NoName** National Taiwan University	**6** 532	3 1 try	45 3 tries			 2 tries	 5 tries	191 2 tries	50 1 try	124 3 tries				19 1 try
13	**HongLongLongLong** National Taiwan Ocean University	**6** 615	8 1 try	43 2 tries	257 1 try	 3 tries	 3 tries	134 2 tries		81 1 try					52 1 try
14	**NCTU_HongLongLong** National Chiao Tung University	**6** 721	37 1 try	33 2 tries			277 4 tries	71 2 tries		152 1 try	 1 try				51 1 try
15	**OwO** National Taiwan University	**6** 723	10 1 try	120 1 try	259 1 try		 1 try	 6 tries		90 1 try	228 1 try				16 1 try
16	**bzpileafd** National Tsing Hua University	**6** 762	13 1 try	53 1 try	 1 try		217 3 tries	276 4 tries		95 1 try					8 1 try
17	**ToBeContinued** National Taiwan University	**6** 801	6 1 try	99 3 tries	207 1 try	 3 tries	 2 tries	279 2 tries		140 1 try					10 1 try
18	**28** National Tsing Hua University	**6** 839	3 1 try	146 4 tries			186 1 try	 4 tries		199 4 tries	141 2 tries		 3 tries		24 1 try
19	**Watame did nothing wrong** National Tsing Hua University	**6** 966	6 1 try	21 2 tries		 2 tries		285 10 tries		89 1 try	231 5 tries				54 1 try
20	**110NTNU** National Taiwan Normal University	**5** 413	4 1 try	30 1 try			 1 try	204 4 tries		100 1 try					15 1 try
21	**NCTU_LoTaTea** National Chiao Tung University	**5** 448	6 1 try	112 1 try			 5 tries	163 2 tries		94 2 tries					33 1 try
22	**NCTU_Eclipse** National Chiao Tung University	**5** 528	5 1 try	81 1 try			 1 try	 3 tries		107 3 tries	234 3 tries				21 1 try
23	**CCU_Senpai Sayounara** National Chung Cheng University	**5** 605	18 1 try	49 1 try	289 3 tries			 2 tries		164 2 tries					25 1 try
24	**NCTU__** National Chiao Tung University	**5** 694	12 1 try	35 1 try			 3 tries	191 2 tries		291 7 tries					25 1 try
25	**NCTU_White** National Chiao Tung University	**4** 131	7 1 try	35 2 tries						59 1 try	 4 tries		 1 try		10 1 try
26	**NTNU_ch1W4w4** National Taiwan Normal University	**4** 180	6 1 try	19 1 try	 1 try			 4 tries		122 2 tries					13 1 try
27	**MCU_Shark** Ming Chuan University	**4** 224	7 1 try	46 1 try			 4 tries			113 1 try					58 1 try
28	**RedThunder** National Taiwan Normal University	**4** 255	14 2 tries	65 1 try				 6 tries		117 1 try	 3 tries				39 1 try
29	**NCTU_Pusheen** National Chiao Tung University	**4** 271	8 1 try	48 2 tries				 3 tries		161 1 try					34 1 try
30	**Rabbit Foot** National Taiwan Normal University	**4** 332	8 1 try	50 1 try	246 1 try					 8 tries					28 1 try
31	**BITreap** National Tsing Hua University	**4** 335	6 1 try	42 1 try	 1 try			 3 tries		171 2 tries	 2 tries				96 1 try
32	**NCU_309** National Central University	**4** 338	35 1 try	101 3 tries	117 1 try					 11 tries					45 1 try
33	**zappers** Fu Jen Catholic University	**4** 347	19 1 try	116 2 tries				 3 tries		161 1 try					31 1 try
34	**HmmmHuhh** National Taiwan Ocean University	**4** 358	18 2 tries	122 1 try		165 1 try	 2 tries								33 1 try
35	**KWS** National Tsing Hua University	**4** 367	6 1 try	55 3 tries	 1 try					210 3 tries					16 1 try

RANK	TEAM	SCORE	PA	PB	PC	PD	PE	PF	PG	PH	PI	PJ	PK	PL	PM
36	**LLC** National Cheng Kung University	**4** 433	16 2 tries	113 1 try						269 1 try					15 1 try
37	**Really Simply Accepted** National Taiwan Normal University	**4** 434	5 1 try	231 3 tries	1 try		1 try			121 2 tries					17 1 try
38	**SHAAAAARK** National Tsing Hua University	**4** 445	7 2 tries	110 2 tries	249 1 try										39 1 try
39	**SAI_sixtieth** National Tsing Hua University	**4** 471	11 1 try	139 4 tries	199 2 tries		5 tries								42 1 try
40	**NUK_4+1BadBad** National University of Kaohsiung	**4** 474	41 1 try	225 1 try	143 1 try										65 1 try
41	**FJCU_Return0** Fu Jen Catholic University	**4** 475	12 2 tries	95 2 tries	2 tries		4 tries	2 tries		242 4 tries					26 1 try
42	**Anti-Accept** National Cheng Kung University	**4** 476	14 1 try	100 1 try			2 tries	1 try		245 4 tries					57 1 try
43	**the gourmet of the desert** National Tsing Hua University	**4** 486	16 4 tries	84 1 try	293 1 try		5 tries								33 1 try
44	**NCU** National Central University	**4** 491	20 1 try	76 3 tries	250 2 tries		2 tries								85 1 try
45	**Acpc** National Taiwan Normal University	**4** 499	20 1 try	124 1 try	267 4 tries										28 1 try
46	**GWAWA** National Central University	**4** 508	10 1 try	132 1 try		260 4 tries				8 tries					46 1 try
47	**NCTU_KokushiMusou** National Chiao Tung University	**4** 535	16 2 tries	124 5 tries	2 tries					243 2 tries					32 1 try
48	**NCU_Ouo** National Central University	**3** 96	9 2 tries	50 1 try	1 try					2 tries					17 1 try
49	**NCTU_Daisangen** National Chiao Tung University	**3** 99	5 1 try	66 1 try						2 tries			1 try		28 1 try
50	**NCTU_Banana** National Chiao Tung University	**3** 105	9 1 try	53 1 try				9 tries		8 tries					23 2 tries
51	**Mine englishes bed** National Taiwan Ocean University	**3** 130	9 1 try	85 1 try	1 try		1 try	8 tries							36 1 try
52	**C++S** National Cheng Kung University	**3** 146	23 2 tries	70 1 try			3 tries	2 tries		3 tries					33 1 try
53	**TCUMI Need Money** Tzu Chi University	**3** 155	7 1 try	110 1 try	1 try	1 try	1 try	1 try	1 try	1 try	1 try	1 try	1 try	1 try	38 1 try
54	**hands0meda1un** Fu Jen Catholic University	**3** 168	21 2 tries	84 1 try	1 try			3 tries							43 1 try
55	**CCU_Snorlax** National Chung Cheng University	**3** 179	14 1 try	100 3 tries		7 tries		6 tries							25 1 try
56	**NTPU_kite** National Taipei University	**3** 180	12 1 try	100 3 tries			6 tries								28 1 try
57	**NCTU_Chinroutou** National Chiao Tung University	**3** 191	17 2 tries	128 1 try											26 1 try
58	**NCTU_3.12B** National Chiao Tung University	**3** 195	9 2 tries	147 1 try			6 tries			9 tries					19 1 try
59	**bits/stdc++.h** Fu Jen Catholic University	**3** 203	14 1 try	146 1 try											43 1 try
60	**NSYSU_Ant** National Sun-Yat-Sen University	**3** 212	13 1 try	116 3 tries						2 tries					43 1 try
61	**UCCU** National Cheng-Chi University	**3** 225	15 1 try	150 2 tries			9 tries					4 tries			40 1 try
62	**Jo-Py** National Taipei University of Business	**3** 238	15 1 try	107 1 try			4 tries								116 1 try
63	**NCU_Coder** National Central University	**3** 247	7 1 try	6 tries	224 1 try										16 1 try
64	**NoRiceNoLife** Feng Chia University	**3** 252	14 1 try	173 2 tries				1 try							45 1 try
65	**NTUT_Kn1ghts** National Taipei University of Technology	**3** 266	15 2 tries	183 2 tries			6 tries			12 tries					28 1 try
66	**MCU_Unicorn** Ming Chuan University	**3** 289	18 1 try	182 2 tries			3 tries								69 1 try
67	**CDJ** Fu Jen Catholic University	**3** 289	12 1 try	245 1 try						8 tries					32 1 try
68	**CCU_hahahaha** National Chung Cheng University	**3** 314	12 1 try				9 tries	3 tries		201 4 tries					41 1 try
69	**NTPU_boomer** National Taipei University	**3** 350	15 2 tries	151 7 tries											44 1 try
70	**NSYSU_Dog** National Sun-Yat-Sen University	**3** 358	21 2 tries	251 1 try											66 1 try
71	**BearHaoEn** National Taitung University	**3** 375	13 2 tries	206 4 tries			3 tries	3 tries							76 1 try
72	**CCU_TOOL_MAN** National Chung Cheng University	**3** 378	30 2 tries	256 1 try		3 tries									72 1 try
73	**SendTreePay** National Taitung University	**3** 383	29 2 tries	179 5 tries											75 1 try
74	**MCU_AC** Ming Chuan University	**3** 384	9 2 tries	193 7 tries			3 tries								42 1 try

RANK	TEAM	SCORE	PA	PB	PC	PD	PE	PF	PG	PH	PI	PJ	PK	PL	PM
75	**IndexO** Yuan Ze University	**3** 389	14 1 try	241 2 tries											74 3 tries
76	**Linzoma** National Tsing Hua University	**3** 405	40 2 tries	232 3 tries											73 1 try
77	**CSDD** Yuan Ze University	**3** 419	10 1 try	224 4 tries		1 try		1 try		1 try					65 4 tries
78	**NUK_EhGanPangolin** National University of Kaohsiung	**3** 424	91 6 tries	157 1 try											76 1 try
79	**NTCU_NONAMETEAM** National Taichung University of Education	**3** 430	11 1 try	284 4 tries											75 1 try
80	**XDDD** National Formosa University	**3** 439	31 3 tries	163 4 tries	4 tries										145 1 try
81	**NTNU_import_magic** National Taiwan Normal University	**2** 36	11 1 try	4 tries			1 try	3 tries							25 1 try
82	**WeWantWin** National Taiwan Ocean University	**2** 36	8 1 try	3 tries				5 tries							28 1 try
83	**bubble-milk-tea** National Kaohsiung University of Science and Technology	**2** 57	13 2 tries	8 tries											24 1 try
84	**floccinaucinihilipilifiCATion** National Taiwan Ocean University	**2** 62	8 1 try	10 tries											54 1 try
85	**MCU_Alpin** Ming Chuan University	**2** 65	8 1 try	4 tries											57 1 try
86	**BMI48** Tatung University	**2** 86	7 1 try		2 tries			6 tries							79 1 try
87	**CHU_MapoTofu** Chung Hua University	**2** 99	30 2 tries	7 tries	1 try										49 1 try
88	**NTNU_WWW** National Taiwan Normal University	**2** 106	22 3 tries	4 tries				8 tries							44 1 try
89	**Crane Game University** Feng Chia University	**2** 122	18 3 tries		3 tries		4 tries								64 1 try
90	**king of coding** National Formosa University	**2** 135	36 2 tries	5 tries			3 tries								79 1 try
91	**no idea** National Chi Nan University	**2** 141	20 2 tries	4 tries											101 1 try
92	**123** Yuan Ze University	**2** 157	25 2 tries				6 tries								112 1 try
93	**IROT** National Kaohsiung University of Science and Technology	**2** 167	29 3 tries				4 tries								78 2 tries
94	**CJCU_AIRC** Chang Jung Christian University	**2** 173	37 4 tries												76 1 try
95	**Soy Milk** National Formosa University	**2** 206	34 3 tries				1 try								132 1 try
96	**NEDuck** National Taipei University of Business	**2** 210	102 4 tries	6 tries			7 tries						3 tries		48 1 try
97	**Three_dogs** Providence University	**2** 212	52 5 tries					1 try							80 1 try
98	**TTU02** Tatung University	**2** 303	46 8 tries							1 try					97 2 tries
99	**TKUECE** Tamkang University	**2** 306	45 1 try	2 tries			3 tries			3 tries					221 3 tries
100	**For The Point** Fu Jen Catholic University	**2** 328	39 1 try												289 1 try
101	**Team Wild Card** National Taipei University of Education	**2** 397	98 5 tries												219 1 try
	SUMMARY	**396**	101 63 0 2min	78 135 0 12min	19 32 0 102min	2 42 0 165min	13 145 0 65min	19 127 0 71min	6 24 0 189min	38 108 0 30min	14 37 0 24min	0 5 0 n/a	5 26 0 180min	0 1 0 n/a	101 11 0 5min

Cell colours
Solved first
Solved
Tried, incorrect
Tried, pending
Untried

Last Update: Sun 08 Nov 2020 16:53:36 CST
using DOMjudge

Problem A
Right-Coupled Numbers

Time limit: 1 second
Memory limit: 1024 megabytes

Problem Description

An integer x is said to be a right-coupled number, if you can find two integers, say $0 < a \leq b \leq x$ such that $a \times b = x$ and $a/b \geq 0.5$. In this problem, your task is to determine whether a given integer is a right-coupled number or not.

Input Format

The first line of the input is an integer N denoting the number of test cases. Each test case is in one line, which contains a single integer $0 < x < 2^{15}$.

Output Format

If the given integer x is a right-coupled number, output 1; otherwise, output 0. Each is in a single line.

Technical Specification

- $1 \leq N \leq 1000$
- $0 < x < 2^{15}$

Sample Input 1

```
4
66
55
105
150
```

Sample Output 1

```
1
0
0
1
```

Almost blank page

Problem B
Make Numbers

Time limit: 1 second

Memory limit: 1024 megabytes

Problem Description

Peter is a math teacher at an elementary school. To familiarize students with three basic arithmetic operations plus (+), minus (−) and times (×), he gives a simple arithmetic puzzle as homework. The puzzle is that you are given 4 digits, and you are told to build as many non-negative integers as possible using just those 4 digits and at least one of the three basic arithmetic operations. For example, you are given 1,1,2,1 as the digits, and then you can build 32 non-negative integers as Table 1.

Table 1: Numbers made by 1,1,2,1.

$0 = 2 - 1 - 1 \times 1$	$22 = 21 + 1 \times 1$
$1 = 2 + 1 - 1 - 1$	$23 = 21 + 1 + 1$
$2 = 2 + 1 - 1 \times 1$	$32 = 21 + 11$
$3 = 2 + 1 + 1 - 1$	$109 = 111 - 2$
$4 = 2 + 1 + 1 \times 1$	$111 = 112 - 1$
$5 = 2 + 1 + 1 + 1$	$112 = 112 \times 1$
$8 = 11 - 2 - 1$	$113 = 112 + 1$
$9 = 11 - 2 \times 1$	$120 = 121 - 1$
$10 = 12 - 1 - 1$	$121 = 121 \times 1$
$11 = 12 - 1 \times 1$	$122 = 121 + 1$
$12 = 12 + 1 - 1$	$132 = 12 \times 11$
$13 = 12 + 1 \times 1$	$210 = 211 - 1$
$14 = 12 + 1 + 1$	$211 = 211 \times 1$
$19 = 21 - 1 - 1$	$212 = 211 + 1$
$20 = 21 - 1 \times 1$	$222 = 111 \times 2$
$21 = 21 + 1 - 1$	$231 = 21 \times 11$

To check whether the student's answer includes all possible integers, Peter needs to know the total number of non-negative integers that can be built for the puzzle. Please write a program to help Peter compute the total number.

Input Format

The input file contains 4 integers separated by a space in a line, which indicates the given digits.

Output Format

Output the total number of non-negative integers that can be built.

Technical Specification

- The expressions are composed by concatenating the 4 given digits and at least one operation in $\{+, -, \times\}$. The given digits are the elements in $\{1, 2, 3, \ldots 9\}$.
- The given digits are partitioned into several groups and the digits in each group are concatenated as a number in arbitrarily permutation order.
- The symbol $-$ can only be treated as a minus operator.
- The operations $+$ and $-$ have equal precedence.
- The operation $\times$ has higher precedence than $+$ and $-$.
- Operations with the highest precedence are evaluated first, and operations with equal precedence are evaluated from left to right.

Sample Input 1

```
1 1 1 1
```

Sample Output 1

```
15
```

Sample Input 2

```
1 1 2 1
```

Sample Output 2

```
32
```

Problem C
Pyramid

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

Consider an $n \times n$ grid where nodes are labelled as (i, j) for $0 \leq i, j < n$. We rotate it 45 degree in clockwise direction and keep only its top half part. Then you get a *pyramid*, as shown in Figure 1. All of the nodes laid on the anti-diagonal of the grid have labels $(n-1-j, j)$ for $0 \leq j < n$. They are located at the bottom line of the pyramid. For simplicity and clarity, we name node $(n-1-j, j)$ as exit j. Node $(0, 0)$ is called the starting point (labelled as P in Figure 1). When you insert a ball from the starting point, this ball will roll down to some of the exits. A ball located at node (i, j) can only roll down to either node $(i+1, j)$ or node $(i, j+1)$, and the ball shall never be broken or split. There is a tiny switch equipped on every node other than the exits that controls the move direction of the ball, and this switch always works well. The switch has exactly two states: either L or R, indicates that the ball can move to node $(i+1, j)$ or $(i, j+1)$, respectively. After the ball leaves this node, the switch changes immediately to the other state. The default setting for a switch is at L.

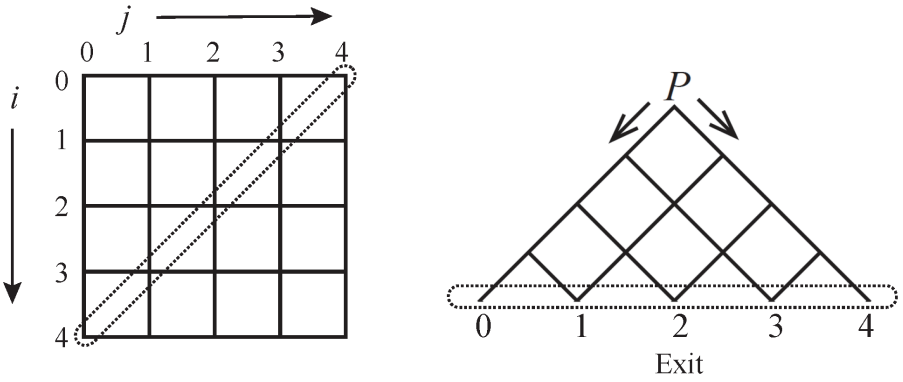


Figure 1: An example for $n = 5$.

When you insert the first ball into P, this ball will reach exit 0, and the states of nodes $(i, 0)$ for $0 \leq i < n-1$ are now all R's. Then you insert the second, third, and so on so forth, one by one, until the k^{th} ball finishes. The status of every switch accumulates with these balls. Please write a program to determine the number of the exit point for the k^{th} ball.

Input Format

The first line of the input is a number that specifies the number of test cases. Each test case has only one line that gives you two space-delimited numbers n and k.

Output Format

Please output the exit number of the k^{th} ball in one line.

Technical Specification

- There are at most 20 test cases.
- $1 \leq n \leq 10^4$.
- $1 \leq k \leq 10^8$.

Sample Input 1

```
2
5 1
5 2
```

Sample Output 1

```
0
1
```

Sample Input 2

```
3
5 3
5 4
5 5
```

Sample Output 2

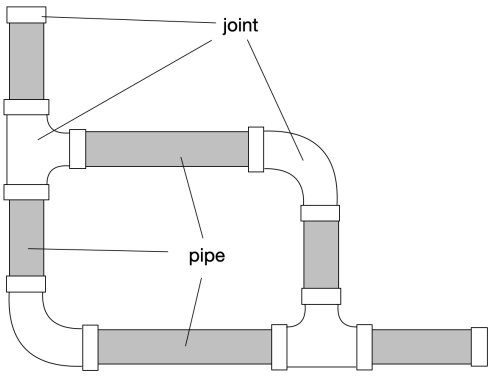
```
2
3
2
```

Problem D
Quality Monitoring

Time limit: 1 second
Memory limit: 1024 megabytes

Problem Description

To provide a better drinking quality, the government is going to deploy some "smart devices" into the water supplying system so that the quality of the water can be monitored. The water supplying system consists of many pipes, and two pipes are connected by a joint. You may assume that the system forms a **connected simple graph**, with pipes being the edges and joints being the vertices. An example is given in the following figure.



The smart devices are designed to be deployed at the joints. However, two adjacent devices may interfere with each other, so it is required that no two devices are adjacent. There have to be enough number of devices deployed so that the system can be **fully monitored**. Formally, the system is fully monitored if

- there are at least $n - 28$ devices deployed, and
- no two devices are adjacent.

Please determine whether the system can be fully monitored. If the answer is no, output -1; otherwise, output the maximum number of devices that can be deployed.

Input Format

The first line of the input file contains two positive integers n and m, where n is the number of joints, numbered from 0 to $n - 1$, and m is the number of pipes. Each of the following m lines contains two nonnegative integers, indicating the joints at two ends of a pipe.

Output Format

Output an integer: "`-1`" if the system cannot be fully monitored; otherwise, the maximum number of devices that can be deployed.

Technical Specification

- $2 \leq n \leq 3 \times 10^4$, $1 \leq m \leq 5 \times 10^5$

Sample Input 1

```
5 7
1 0
2 3
1 4
1 2
3 1
3 4
0 4
```

Sample Output 1

```
2
```

Problem E
A Color Game

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

Playing games is fun. For programmers, however, playing games with programs is even more fun. Consider a simple single-user tabletop game as follows. Given a row of sticks, each of which is in one of the seven colors, red (`R`), green (`G`), blue (`B`), cyan (`C`), magenta (`M`), yellow (`Y`), and key (`K`), the goal of the game is to eliminate all the sticks by repeating the following rules.

- Consecutive sticks with the same color can be eliminated if the size of them is not less than m.
- The remaining sticks will move closer together.

For the case where the row is `BBBRRRRRRGGGB` and m is 3, all the sticks can be successfully eliminated as the following steps:

1. `BBBRRRRRRGGGB`
2. `BBBGGGB` (By eliminating all red sticks)
3. `BBBB` (By eliminating all green sticks)
4. (By eliminating all blue sticks)

For the same row of sticks with $m = 4$, however, it is no way to eliminate all the sticks.

Given a row of n sticks and the value of m, your task is to determine if it is possible to eliminate all the sticks.

Input Format

Each test case is given as a string that is the row of sticks and an integer m.

Output Format

Output `Yes` if it is possible to eliminate all the sticks. Otherwise, output `No`.

Technical Specification

- $0 < n, m \leq 500$

Sample Input 1

```
BBBRRRRRRGGGB 3
```

Sample Output 1

```
Yes
```

Sample Input 2

```
BBBRRRRRRGGGB 4
```

Sample Output 2

```
No
```

Problem F
Cable Protection

Time limit: 2 seconds

Memory limit: 1024 megabytes

Problem Description

A company ICPC (International Cable Protection Company) produces a cable protection tool that can be installed in a network switch to monitor whether all cable links connected to it are working properly. Because the protection tool would cause transmission delay, it is not suitable for installation on every switch.

Usually network topology consists of two parts: a backbone and several subnets. The switches on the backbone are linked as a ring structure and each backbone switch is treated as a root of a subnet in which the switches are linked as a tree structure. We call such topology as unicyclic topology. Figure 2 shows an example of a unicyclic topology.

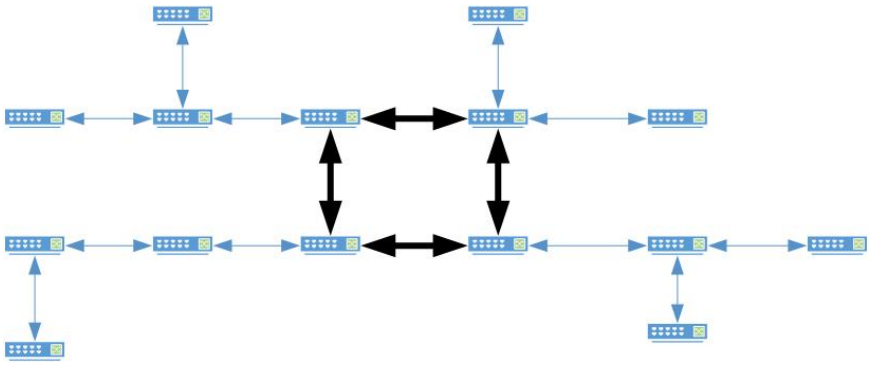


Figure 2: An example of uncyclic topology.

Suppose there are n backbone switches and m subnet switches. The switches are numbered by integers from 0 to $m + n - 1$. Backbone switches are numbered from 0 to $n - 1$ in clockwise order and the subnet switches are numbered from n to $n + m - 1$ where the index of each subnet switch is larger than the index of its parent in the rooted tree structure of the subnet it belongs. Figure 3 shows an example of switch numbering.

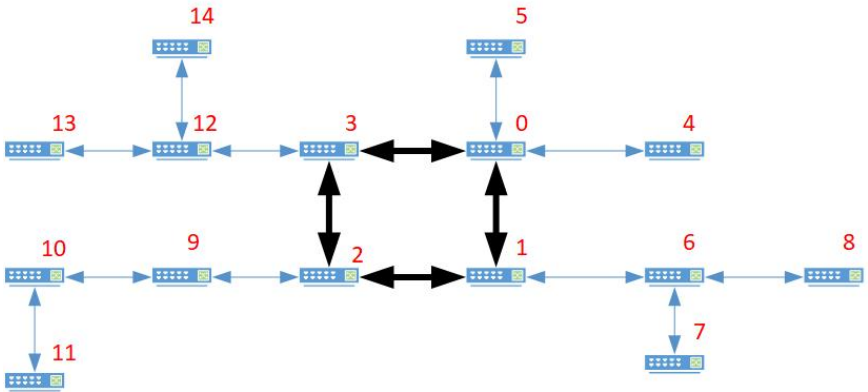


Figure 3: An example of switch numbering.

Please write a program for ICPC to decide the minimum number of switches selected for installing cable protection tools that can monitor all the cable links. Figure 4 shows an optimum solution (circled by ellipses) for the given network.

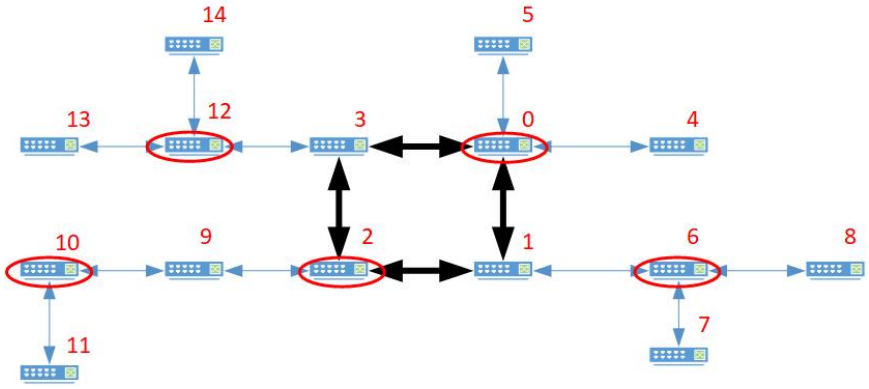


Figure 4: An optimum solution for the given network.

Input Format

The first line of the input file contains two integers n and m, separated by a space, indicating the numbers of backbone switches and subnet switches respectively. Each of the next $n+m$ lines consists of two integers, separated by a space, indicating the indices of the two end switches of a link.

Output Format

Output the minimum number of switches selected for installing cable protection tools that can monitor all the cable links.

Technical Specification

- $3 \leq n \leq 100000$
- $1 \leq m \leq 100000$

Sample Input 1

```
3 2
0 1
1 2
0 2
1 3
2 4
```

Sample Output 1

```
2
```

Sample Input 2

```
4 11
0 1
0 3
0 4
0 5
1 2
1 6
2 3
2 9
3 12
6 7
6 8
9 10
10 11
12 13
12 14
```

Sample Output 2

```
5
```

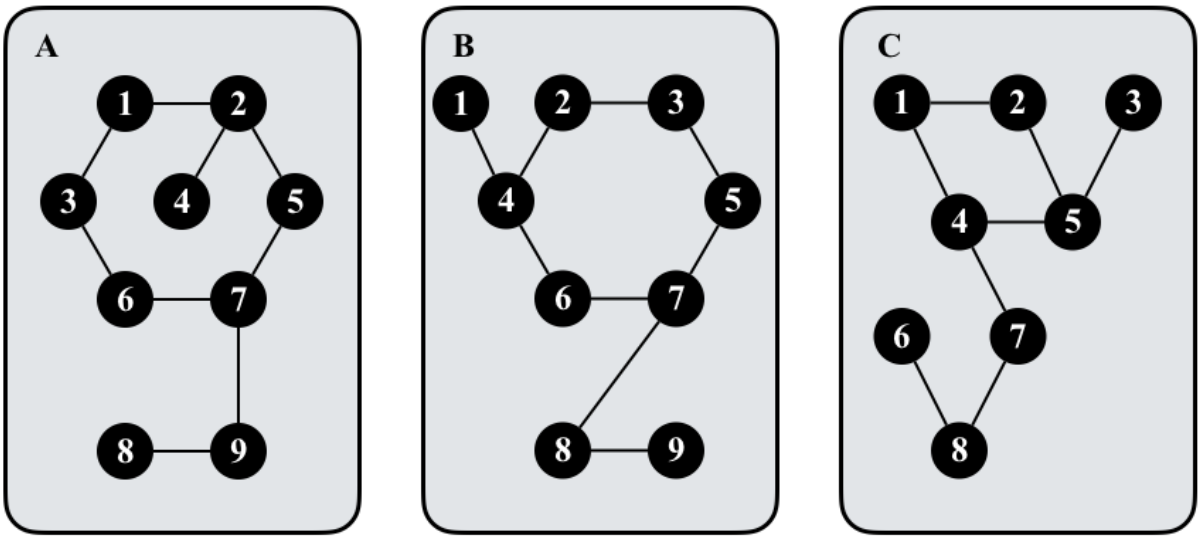
Almost blank page

Problem G
Graph Cards

Time limit: 30 seconds
Memory limit: 1024 megabytes

Problem Description

A deck of graph cards is placed on the table. Each graph card χ is decorated with an undirected simple graph G_χ so that G_χ is connected and G_χ has the same number of nodes and edges. Note that different graph cards may have different numbers of nodes. An example is depicted as follows.



We say two graph cards are identical if and only if the graphs associated with them, say $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, are ***isomorphism***; that is, there exists a bijection f between the node sets V_1 and V_2 so that for every $x, y \in V_1$, edge $(x, y) \in E_1$ if and only if edge $(f(x), f(y)) \in E_2$. Our goal is to compute the number of distinct graph cards in the deck.

Input Format

The first line contains an integer t that indicates the number of test cases. For each test case, you are given a deck of graph cards. It begins with a line containing the number of graph cards $n > 0$. Then, n lines follow. Each line represents a graph card associated with a graph G in the following format:

$$k \;\; u_1 \;\; v_1 \;\; u_2 \;\; v_2 \;\; \cdots \;\; u_k \;\; v_k$$

where $k > 0$ denotes the number of nodes (also edges) in G and for each $i \in [1, k]$ (u_i, v_i) denotes an edge in G that connects nodes u_i and v_i. Note that the identifiers of nodes are integers in $[1, k]$.

Output Format

For each test case, output the number of distinct graph cards in the given deck on a line.

Technical Specification

- $0 < t \leq 30$.
- $0 < n, k \leq 10^6$.
- For each test case, the numbers of nodes in the n graph cards sum up to at most 10^6.

Sample Input 1

```
1
2
4 1 2 2 3 3 1 1 4
4 1 2 2 3 3 1 2 4
```

Sample Output 1

```
1
```

Sample Input 2

```
2
2
4 1 2 2 3 3 1 1 4
5 1 2 2 3 3 1 2 4 2 5
3
9 1 2 2 5 5 7 7 6 6 3 3 1 2 4 7 9 9 8
9 1 4 4 2 2 3 3 5 5 7 7 6 6 4 7 8 8 9
9 1 2 2 5 5 4 4 1 4 7 7 8 8 6 8 9 5 3
```

Sample Output 2

```
2
2
```

Problem H
Optimization for UltraNet

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

The UltraNet company provides network connectivity for all cities in a country. For a pair of cities, they are either directly connected or indirectly connected. A city i and another city j are directly connected if a cable with a symmetrical bandwidth of $b_{i,j} = b_{j,i}$ is wired between them. For two cities that are not directly connected, at least one path between the two cities exists. In the current UltraNet, more than one path is possibly available for a city pair.

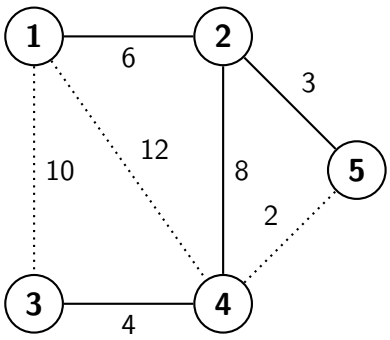
The current UltraNet is perfectly working, while the maintenance fee of each cable is costly. Energy conservation is another concern. The energy consumption of a cable is proportional to its bandwidth. Therefore, the company has an optimization plan to reorganize the network with the policies in the following order:

1. The number of cables should be minimized without sacrificing the connectivity of the whole UltraNet. In other words, exactly one path between every city pair should be satisfied.

2. If there is more than one way to minimize the number of cables, the bottleneck of the whole network should be maximized. The bottleneck of a network is determined by the city pair with the narrowest bandwidth, and the bandwidth of a city pair (i, j), $b'_{i,j}$, is determined by the cable with the narrowest bandwidth on the only path from i to j.

3. If there is still more than one way to meet the above two points, the total energy consumption of the network should be minimized. In other words, the sum of bandwidths of the remaining cables should be minimized.

Your task is to help UltraNet optimize the network and then output the sum of the bandwidths among all city pairs in the optimized network. For optimizing the following example, the three cables in dotted will be discarded. In the resulting network, the bottleneck is 3, the sum of bandwidths of the remaining four cables is $6 + 3 + 8 + 4 = 21$, and the sum of the bandwidths among all city pairs is $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} b'_{i,j} = 6 + 4 + 6 + 3 + 4 + 8 + 3 + 4 + 3 + 3 = 44$.



Input Format

Each test case begins with two integers n and m, denoting numbers of cities and cables, respectively. Then, m lines will follow for specifying the m cables. Each of the m lines contains three positive integers, i, j, and $b_{i,j}$, denoting that a cable with a bandwidth of $b_{i,j}$ directly connects the city pair (i, j), where the cities are numbered from 1 to n, and $i < j$ since $b_{i,j} = b_{j,i}$. No more than one cable will be available between every city pair in the current network. In addition, the bandwidths of all cables are distinct; no two cables have the same bandwidth rating.

Output Format

The output is a single integer that is the sum of the bandwidths of all city pairs in the optimized network.

Technical Specification

- $2 \leq n \leq 10000$
- $1 \leq m \leq 500000$
- $1 \leq i < j \leq n$
- $0 < b_{i,j} < 10^7$

Sample Input 1

```
3 3
1 2 5
1 3 6
2 3 8
```

Sample Output 1

```
20
```

Sample Input 2

```
5 7
1 2 6
1 3 10
1 4 12
2 4 8
2 5 3
3 4 4
4 5 2
```

Sample Output 2

```
44
```

Sample Input 3

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

Sample Output 3

```
24
```

Almost blank page

Problem I
Critical Structures

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

Intelligence Cloud Privacy Company (ICPC) is a world famous cloud service company that aims at developing secure and powerful cloud computing environments for users. Engineers in the ICPC construct a data center with n computing nodes, denoted by $1, 2, \ldots, n$, and m communication links. We can model this data center as an undirected graph $G = (V, E)$, in which n vertices represent n computing nodes and an edge between Node i to Node j if there is a communication link between them; we also call i and j are two end-nodes of this link. In addition, for two arbitrary nodes i and j in G, there is at most one communication link between i and j, and there is no communication link between the same node.

A *linear array structure* in a data center G is a sequence of nodes $\langle v_0, v_1, \ldots, v_{k-1} \rangle$, where $k \geq 2$, such that any two consecutive v_{i-1} and v_i for $1 \leq i \leq k-1$ have a communication link, and v_i for $0 \leq i \leq k-1$ are all distinct. A *ring structure* is a sequence of nodes $\langle v_0, v_1, \ldots, v_{k-1} \rangle$, where $k \geq 4$, such that any two consecutive v_{i-1} and v_i for $1 \leq i \leq k-1$ have a communication link, $v_0 = v_{k-1}$ and v_i for $0 \leq i \leq k-2$ are all distinct. A data center G is *connected* if there is a linear array between any two nodes; otherwise, it is *disconnected*. Using some elegant resource allocation algorithm, a research team of the ICPC needs to find the following critical structures for enhancing the privacy and security:

1. Critical node: a node in G whose removal disconnects G.
2. Critical link: a communication link in G whose removal disconnects G.
3. Critical component: a maximal set of communication links in G such that any two communication links in the set lie on a common ring.
4. Largest critical component: a critical component with the maximum number of communication links.

Given a connected data center G, your task is to write a computer program for computing the number of critical nodes, the number of critical links, and

$$\begin{aligned} \mu^* &= \frac{\text{the number of critical components}}{\text{the number of communication links in a largest critical component}} \\ &= \frac{p}{q}, \end{aligned}$$

where $\frac{p}{q}$ is an irreducible form of μ^*.

Input Format

The first line of the input file contains an integer L ($L \leq 10$) that indicates the number of test cases as follows. For each test case, the first line contains two integers (separated by a space) representing n and m. Then it is immediately followed by m lines, in which each line contains two integers that represent two end-nodes of a communication link; moreover, any two consecutive integers are separated by a space.

Output Format

The output contains one line for each test case. Each line contains four positive integers representing the number of critical nodes, the number of critical links, p, and q, where $\frac{p}{q}$ is an irreducible form of μ^*. Note that any two consecutive integers are separated by a space.

Technical Specification

- $3 \leq n \leq 1000$ for each test case.
- $n - 1 \leq m \leq \frac{n(n-1)}{2}$.
- The sum of m in all L tests is smaller than 10^6.

Sample Input 1

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

Sample Output 1

```
0 0 1 6
```

Sample Input 2

```
1
6 7
1 2
2 3
3 1
4 5
5 6
6 4
1 4
```

Sample Output 2

```
2 1 1 1
```

Almost blank page

Problem J
Puzzle Game

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

For a string S, define $Adjacency(S)$ to be the *multiset* of *unordered* pairs $(S[i], S[i+1]), i = 1, 2, ..., |S| - 1$, and define $\Sigma(S)$ to be the multiset of $S[i], i = 1, 2, ..., |S|$, where $|S|$ is the length of S and $S[i]$ is the ith character of S. For example, for S = ABADDADCAB, we have $Adjacency(S)$ = {AB, BA, AD, DD, DA, AD, DC, CA, AB} = {AB, AB, AB, AC, AD, AD, AD, CD, DD} and $\Sigma(S)$ = {A, A, A, A, B, B, C, D, D, D}.

John is playing a puzzle game, in which two strings P and Q, $|P| > |Q|$, over the character set {A, B, C, D} are given and the goal is to insert characters into Q to obtain a string Q' such that $\Sigma(Q') = \Sigma(P)$ and $Adjacency(Q') = Adjacency(P)$. For example, given P = ABADCAB and Q = CBB, by inserting A, D, A, A into Q, we can obtain a string Q' = ADCABAB, in which inserted characters are underlined. It is easy to check that $\Sigma(Q') = \Sigma(P)$ = {A, A, A, B, B, C, D} and $Adjacency(Q') = Adjacency(P)$ = {AB, AB, AB, AC, AD, CD}. Thus, Q' is a solution for P = ABADCAB and Q = CBB. As another example, for P = ABA and Q = CB, there is no solution.

Please write a program to help John. More specifically, given two strings P and Q, your program computes a string Q' such that Q' is obtained from Q by inserting some characters, $\Sigma(Q') = \Sigma(P)$, and $Adjacency(Q') = Adjacency(P)$.

Input Format

The first line of the input is an integer t, indicating that there are t test cases. Each test case consists of three lines: the first gives two integers, indicating the lengths $|P|$ and $|Q|$, the second gives the string P, and the third gives the string Q.

Output Format

For each case, output a solution string Q'. If there are multiple solutions, you can output any of them. If there is no solution, output "NO".

Technical Specification

- The number of test cases is at most 15.
- The length of P, $|P|$, is an integer between 2 and 10^3.
- The length of Q, $|Q|$, is an integer between 1 and 10^3 and $|P| - 18 \leq |Q| \leq |P| - 1$.
- Both P and Q are over the character set {A, B, C, D}.

Sample Input 1

```
3
7 3
ABADCAB
CBB
11 7
ABACCDBADAC
AADCDAC
3 2
ABA
CB
```

Sample Output 1

```
ADCABAB
ABABDCCADAC
NO
```

Problem K
Number with Bachelors

Time limit: 2 seconds

Memory limit: 1024 megabytes

Problem Description

Numbers without duplicated digits are considered bachelor numbers. For example, 123 is a bachelor number in decimal number system, and 9af is a bachelor number in a hexadecimal number system. Both decimal number 101 and hexadecimal number aba are not bachelor numbers since there are duplicated digits in them. In this problem, you get two types of question. For one, given an interval, say, $[a, b]$ in a designated number system, decimal or hexadecimal, you have to figure out the total number of bachelor numbers in the interval, including a and b. For another, you are given a number, say, i in a designated number system you have to find the i^{th} bachelor number in that number system.

Input Format

The first line of the input is a number n, which specifies the number of test cases. Each test case is a question and appears in one line. Each question starts with a letter 'd' or 'h', indicating the question is in decimal domain or hexadecimal domain, respectively. For decimal domain, the following numbers are all represented in decimal. For hexadecimal domain, the following numbers are all represented in hexadecimal. Next to the first letter is a digit 0 or 1, indicating the type of question to be asked. For type 0 question, two integers a and b ($0 \leq a \leq b < 2^{64}$) follow, which represent an interval. For type 1 question, an integer $1 \leq i < 2^{64}$ follows, which represents an order.

Output Format

Output an integer for each question in its corresponding test case. For each question in decimal domain, the answer must be in decimal. For each question in hexadecimal domain, the answer must be in hexadecimal. For type 1 question, if the i^{th} bachelor number does not exist, output a single letter '-' in its corresponding line.

Technical Specification

- $1 \leq n \leq 50000$.
- $0 \leq a \leq b < 2^{64}$.
- $1 \leq i < 2^{64}$.

Sample Input 1

```
6
d 0 10 20
h 0 10 1f
d 1 10
h 1 f
d 1 1000000000
h 1 ffffffffffffffff
```

Sample Output 1

```
10
f
9
e
-
-
```

Problem L
Save lives or money

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

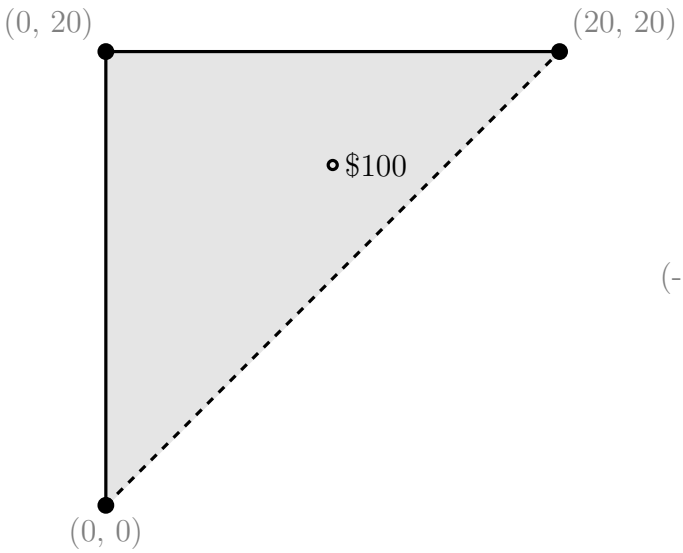
Village "Under The Sea" is located inside a valley. There is a big river in front of the only entry of the village. This year, they encounter a flood that happens roughly once in a century. Because the government is lack of awareness, it is too late to evacuate the residents. The water will flow into the village soon.

Fortunately, this village has walls and gates that could block the water. But we cannot block all the water outside. Otherwise there will be too much water flowing through the river and destroy a nuclear plant in a neighborhood of the village, and brings incalculable damage to everyone. We need to allow some water flowing in, with a manageable way.

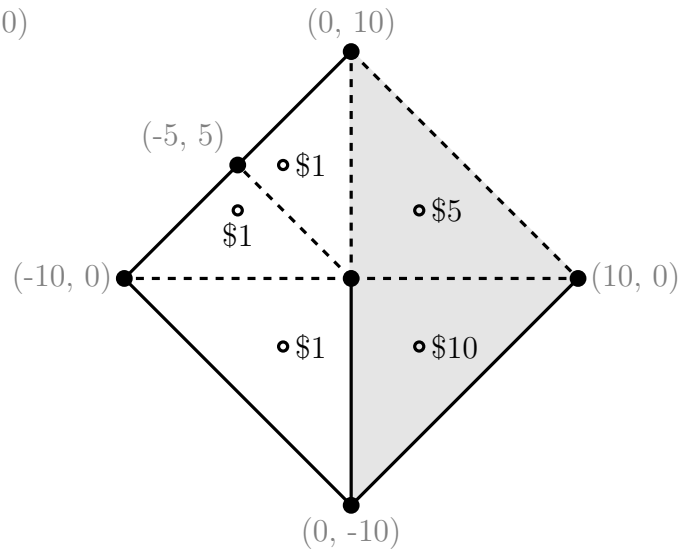
The walls and gates separate the village into many closed regions. Any two different regions could reach each other with exactly one path through the gates if we open all of them. To be clear, the sample 1 is a village consists of 1 region with 2 walls and 1 gate. The solid lines are walls and the dashed line is a gate in the figure below. And the sample 2 is another village consists of 5 regions with 5 walls and 5 gates. Given the estimated water volume, the government could decide to close some gates and leave the rest open. Let the floodwater destroy some regions and leave others safe. The shaded regions in the figures are destroyed regions of the best plans in the sample outputs.

A government official asks you to write a program to help them choosing which gates to open. They give you a list consisted of all the residents with the place they live and money they own. The government official wants you to find a way to save people with the most total wealth. You feel not good to treat rich and poor people differently. So you want to do something different in secret. You will give a plan which save the most people instead. In case there are different plans that save the same number of people, then you choose the one that saves the most money among them.

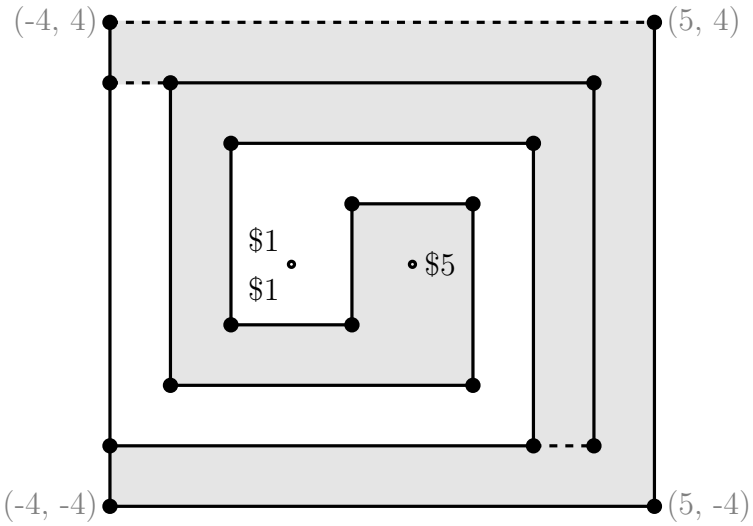
Sample 1



Sample 2



Sample 3



Input Format

The first line contains 1 integer *Area* – the estimated area that the flood will destroy.

The second line contains 3 integers G, W, and R – the number of the gates, walls, and the residents.

Then G lines follow. Each line contains 4 integers x_{1_g}, y_{1_g}, x_{2_g}, y_{2_g} that represent the coordinates of the two endpoints of a gate.

Then W lines follow. Each line contains 4 integers x_{1_w}, y_{1_w}, x_{2_w}, y_{2_w} that represent the coordinates of the two endpoints of a wall.

Finally, there are R lines. Each line contains 3 integers x_r, y_r, and $money_r$ that represent the coordinates of a resident and the amount of money they owns.

Output Format

You should output 2 lines.

The first line has 1 real number and then 3 integers *area*, *money*, *people*, and *gate_n*, which represent the result of the plan. *area* is a real number rounding to the nearest tenth after the decimal point, which is the total area of destroyed regions. *money* is the total amount of money of the victims. *people* is the number of the victims. *gate_n* is the number of the opened gates.

The second line has *gate_n* integers which are the indices of the opened gates in an arbitrary order. Note that the gates are indexed from 1 to G.

If the *Area* in the input is larger than the village, the *area* you output should be the whole size of the village, the *money* should be the total amount of money of all the people in the village, and the *people* should be all the people in the village. And you should open all the gates.

If the *Area* in the input is no more than the village, the *area* you output should be equal to or larger than *Area*.

If there are multiple solutions that can save the same number of people, choose the one which loses less money. If there are still multiple solutions which lose the same amount of money, choose the one with smaller destroyed area. If there are still multiple solutions which destroy the same size of area, anyone will do.

Technical Specification

- $0 < area, G, W, R < 5000$
- $-5000 < x, y < 5000$
- $0 \leq money < 5000$
- There is exactly one gate on the boundary of the village. The water will flood into the village through this gate. This gate should be opened in a workable plan.
- All the regions are simple polygons. They do not intersect themselves and have no holes.
- All the walls or gates will not intersect with each other. They will touch others only at the endpoints.
- Each endpoint will connect to at least two walls or gates. There is no hanging wall or gate.
- All the positions of the residents will locate in the interior of regions. They will not be outside of the village. And they will not sit right on a wall, a gate, nor a junction.

Sample Input 1

```
20
1 2 1
0 0 20 20
20 20 0 20
0 20 0 0
10 15 100
```

Sample Output 1

```
200.0 100 1 1
1
```

Sample Input 2

```
100
5 5 5
0 10 10 0
0 0 0 10
0 0 10 0
0 0 -10 0
0 0 -5 5
0 -10 -10 0
-10 0 -5 5
0 10 -5 5
10 0 0 -10
0 0 0 -10
3 3 5
-5 3 1
-3 5 1
-3 -3 1
3 -3 10
```

Sample Output 2

```
100.0 15 2 2
1 3
```

Sample Input 3

```
33
3 17 3
-4 4 5 4
-4 3 -3 3
3 -3 4 -3
0 1 0 -1
-4 3 -4 -3
-3 -2 -3 3
-2 2 -2 -1
2 1 2 -2
3 2 3 -3
4 3 4 -3
-3 3 4 3
-2 2 3 2
-2 -1 0 -1
0 1 2 1
-3 -2 2 -2
-4 -3 3 -3
-4 -4 5 -4
-4 -4 -4 -3
-4 3 -4 4
5 -4 5 4
1 0 5
-1 0 1
-1 0 1
```

Sample Output 3

```
48.0 5 1 2
1 3
```

Almost blank page

Problem M
Keystroke

Time limit: 1 second
Memory limit: 1024 megabytes

Problem Description

You are designing a numeric keypad for numbers 1 to 4, where each number is associated with a unique key. All of the keys are arranged as a 2×2 matrix, and there is a detection circuit beneath the keypad. When a key is pressed, the circuit will transmit the keystroke signals to the controller, which will receive its row number and column number. We can use a pair (`row`, `column`) to represent an event of a keystroke. Precisely speaking, when you press the key of number i where $i \in \{1, 2, 3, 4\}$, the controller will receive the pair $(\lfloor (i-1)/2 \rfloor, (i-1) \bmod 2)$. For example, when you press key 3, the controller gets $(1, 0)$ as the keystroke signal. You would like to press several keys at the same time for some reason. When you do this, the controller can still receive their corresponding row/column numbers. However, their row numbers are mixed together, as well as the column numbers. For example, when you press keys 1 and 4 simultaneously, the controller would get row numbers $\{0, 1\}$ and column numbers $\{0, 1\}$, because key 1 emits $(0, 0)$ and key 4 emits $(1, 1)$. Another example is that when you pressed keys 1 and 2 simultaneously, the controller can only receive $(\{0\}, \{0, 1\})$ because key 1 emits $(0, 0)$ and key 2 emits $(0, 1)$ and their row numbers are the same. Notice that different keystroke combinations may lead to the same signal. Press keys 2 and 3 would get $(\{0, 1\}, \{0, 1\})$ which is identical to press 1 and 4. Press keys $1, 2, 3, 4$ simultaneously would get the same result. Given a keystroke signal, which is in the (`row`, `column`)-paired form, please write a program to identify the total number of possible keystroke combinations that can lead to this signal.

Input Format

The first line of the input is a positive integer that specifies the number of test cases. Each test case follows immediately in the next line after the previous one. In each test case, its first line gives you two positive integers m and n. Its second line gives you m distinct integers that are the received row numbers. Its third line gives you n distinct integers that are the received column numbers. All numbers in the same line are space-delimited.

Output Format

Output the result in a single line for each test case.

Technical Specification

- There are at most 10 test cases.
- $1 \leq m, n \leq 2$.

Sample Input 1

```
2
2 1
0 1
0
1 2
1
0 1
```

Sample Output 1

```
1
1
```

Sample Input 2

```
2
2 2
0 1
0 1
1 1
1
1
```

Sample Output 2

```
7
1
```