The 19th Zhejiang University Programming Contest

Sponsored by

图森 tu Simple

Contest Session

April 14, 2019

Problem List

A	Thanks, TuSimple!
B	Even Number Theory
C	Robot Cleaner I
D	Robot Cleaner II
E	Potion
F	Strange Function
G	Postman
H	Rescue the Princess
I	Defense Plan
J	Extended Twin Composite Number

This problem set should contain 10 (ten) problems on 13 (thirteen) numbered pages. Please inform a runner immediately if something is missing from your problem set.

Prepared by SUA Programming Contest Problem Setter Team.
https://sua.ac/

Hosted by

Sponsored by

Problem Set Prepared by

It's against the rules to open non-contest websites during the contest.
If you're interested (which is our pleasure), please scan
the QR code only after the contest.

Problem A. Thanks, TuSimple!

In the very first sentence of the very first problem, we would like to give our sincere thanks to TuSimple, the sponsor of this contest.

Founded in 2015, TuSimple is a global autonomous trucking solution company headquartered in San Diego, operating self-driving trucks out of Tucson, Arizona. TuSimple is now developing a commercial-ready Level 4 (SAE) fully-autonomous driving solution for the logistics industry. TuSimple's trucks are the first and only capable of self-driving from depot-to-depot and do so every day for its customers. The company is driven by a mission to increase safety, decrease transportation costs, and reduce carbon emissions.

Nowadays, the trucking industry is currently facing a shortage of 50000 drivers (which is expected to increase to 175000 by the end of 2024) and is approaching a 100 percent turnover rate per year with an average driver age of 49 years old. According to a PwC study, autonomous trucking technologies will reduce annual operating costs for a traditional average long-haul truck by 28 percent in 2025. TuSimple is aiming to transform the 740-billion U.S. trucking industry by cutting costs, reducing carbon emissions and eradicating some of the challenges currently faced by operators.

Building the industry's first 1000-meter perception system, TuSimple soon becomes the pioneer in the industry. As is known to us, 1000 meters can provide 35 seconds of reaction time on average at highway speeds, enabling the system to make the safest and most efficient driving decisions. Even in the adverse weather conditions, the perception system is still designed to identify objects and obstacles, ensuring the safety of both cargoes, trucks, and passers-by. On the other hand, TuSimple's latest proprietary AI is now capable of long-distance highway driving and complex surface street driving, which allows fully autonomous deliveries from one depot to another.

Despite their advanced technology and an enormous sense of mission in the industry, TuSimple shares the corporate culture of honesty, realistic, exploration and innovation among their employees from bottom to top, which allows them to attract more and more elites from all expertise to join and get involved. 'Here's why a little-known autonomous trucking company is beating Tesla and Waymo in the race for driverless big rigs', commented by Business Insider.

The future of trucking is now!

As a manager of TuSimple, you are going to hold a dancing party for both the Development Department and the Marketing Department. There will be n gentlemen and m ladies in total and they are going to dance in pairs. After a careful investigation, we have already known that for each person, they like to dance with either a taller person or a person with smaller height. To simplify the problem, there are no two persons of the same height and people are only allowed to dance with a person of the opposite gender. In order to reserve a proper dancing field, you must calculate the maximum possible number of pairs of people dancing at the same time.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n, m ($1 \le n, m \le 10^5$), indicating the number of gentlemen and ladies.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \le a_i \le 2^{31} - 1$, $a_i \ne a_j$ for all $i \ne j$), indicating the height of each gentleman.

The third line contains m integers $b_1, b_2, \dots, b_m$ ($0 \le b_i \le 2^{31} - 1$, $b_i \ne b_j$ for all $i \ne j$), indicating the height of each lady.

The fourth line contains n integers $p_1, p_2, \dots, p_n$ ($0 \le p_i \le 1$), indicating the preference of each gentleman. If $p_i = 0$, it means gentleman i prefers to dance with a lady of smaller height. Otherwise it indicates he prefers to have a taller dancing partner.

The fifth line contains m integers $q_1, q_2, \dots, q_m$ ($0 \le q_i \le 1$), indicating the preference of each lady. If $q_i = 0$, it means lady i prefers to dance with a gentleman of smaller height. Otherwise it indicates she prefers to have a taller dancing partner.

It's guaranteed that the sum of n and m of all test cases will not exceed 10^6 and $a_i \ne b_j$ for all $1 \le i \le n, 1 \le j \le m$.

Output

For each test case output one line containing one integer, indicating the answer.

Example

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

Note

In the sample test case, the 1st gentleman can dance with the 2nd lady, and the 2nd gentleman can dance with the 1st lady.

Problem B. Even Number Theory

Let $\mathbb{E} = \{2k | k \in \mathbb{Z}^+\}$, which is the set of all positive even numbers. Define the following concepts:

- E-prime: A positive even number p is an e-prime, if and only if there does not exist two integers a and b, such that $a, b \in \mathbb{E}$ and $p = ab$. For example, 2 and 18 are e-primes, but 16 is not, as $16 = 2 \times 8$.
- E-prime factorization: An e-prime factorization of a positive even number e is the decomposition of e into the product of some smaller e-primes.

  More formally, an e-prime factorization of a positive even number e is a **multiset** (a set which allows duplicated elements) $\mathbb{P}$ such that

  - For all $p \in \mathbb{P}$, p is an e-prime;
  - $\prod_{p \in \mathbb{P}} p = e$.

  Please note that, different from the traditional number theory, the e-prime factorization of a positive even number e is NOT unique. For example, we can factorize 36 into 2×18 or 6×6.
- E-factorial: Let $e!!$ be the e-factorial of a positive even number e, we have

$$e!! = \prod_{k \in \mathbb{E}, k \leq e} k$$

  For example, $6!! = 2 \times 4 \times 6 = 48$.

Given a positive even number e, your task is to find an e-prime factorization $\mathbb{P}$ of $e!!$, such that $|\mathbb{P}|$ (the size of $\mathbb{P}$) is as large as possible. In order to make the task easier, you just need to output the value of $|\mathbb{P}|$.

Input

There are multiple test cases. The first line of the input contains an integer T (about 50), indicating the number of test cases. For each test case:

The first and only line contains a positive even number e ($2 \leq e \leq 10^{1000}$), indicating the given number.

Output

For each test case output one integer in one line, indicating the maximum size of the e-prime factorization of $e!!$.

Example

standard input	standard output
2 2 4	1 3

Note

For the first sample test case, as $2!! = 2$ is an e-prime, the answer is (obviously) 1.

For the second sample test case, we can factorize $4!! = 8$ into $2 \times 2 \times 2$, which contains 3 e-primes.

Problem C. Robot Cleaner I

Tired of tidying up his room, BaoBao decides to invent a robot cleaner to help him do the cleaning.

Let's consider BaoBao's room as a grid with n rows and m columns, where the cell in the i-th row and the j-th column is denoted as (i, j). Each cell belongs to one of the following three types:

- Type 0: This cell is empty;
- Type 1: This cell is a wall;
- Type 2: This cell contains a piece of litter.

As we know, rooms are surrounded by walls. So it's guaranteed that for all $1 \le i \le n$, both $(i, 1)$ and (i, m) are walls; And for all $1 \le j \le m$, both $(1, j)$ and (n, j) are walls.

After days of hard work, BaoBao successfully creates his very first robot cleaner. The robot is equipped with five sensors, four wheels, a robot arm, and a very simple controller.

The sensors can detect the type of the cell the robot is currently in, as well as the types of the four neighboring cells. That is to say, if the robot is in cell (i, j), it can know the types of five cells: (i, j), $(i-1, j)$, $(i+1, j)$, $(i, j-1)$ and $(i, j+1)$.

The controller accepts a string s consisting of exactly 243 ($243 = 3^5$) characters as the program, where each character represents an instruction, and controls the robot according to the program and the values returned from the sensors. We now list the valid instructions below, and we assume that the robot is currently in cell (i, j).

- U: If $(i-1, j)$ is not a wall, move the robot from (i, j) to $(i-1, j)$. Otherwise do nothing;
- D: If $(i+1, j)$ is not a wall, move the robot from (i, j) to $(i+1, j)$. Otherwise do nothing;
- L: If $(i, j-1)$ is not a wall, move the robot from (i, j) to $(i, j-1)$. Otherwise do nothing;
- R: If $(i, j+1)$ is not a wall, move the robot from (i, j) to $(i, j+1)$. Otherwise do nothing;
- P: If (i, j) contains a piece of litter, the robot will pick up the litter. Otherwise do nothing. Note that after picking up the litter, (i, j) becomes an empty cell;
- I: Do nothing.

The controller works as follows. Note that we still assume that the robot is currently in cell (i, j), and we denote $t(i, j)$ as the type of cell (i, j).

1. Calculate $x = 3^4 \times t(i, j) + 3^3 \times t(i-1, j) + 3^2 \times t(i+1, j) + 3 \times t(i, j-1) + t(i, j+1)$;
2. Read the $(x+1)$-th character in s as the instruction and execute it. After that, go back to step 1.

Given the map of BaoBao's room, the program string and the starting position of the robot, please calculate the number of pieces of litter the robot can pick up after executing k instructions.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n, m ($n, m \ge 3$, $n \times m \le 2000$), indicating the size of the room.

The second line contains three integers a, b and k ($1 < a < n$, $1 < b < m$, $1 \le k \le 10^{18}$), where a and b indicates that the robot starts from cell (a, b), and k indicates the number of instructions the robot executes.

The third line contains a string s ($|s| = 243 = 3^5$), indicating the program fed into the controller.

For the following n lines, the i-th line contains a string r_i ($|r_i| = m$) consisting of '0', '1' and '2', where the j-th character of r_i indicates the type of cell (i, j).

It's guaranteed that

- For all $1 \le i \le n$, both $(i, 1)$ and (i, m) are walls; And for all $1 \le j \le m$, both $(1, j)$ and (n, j) are walls;
- Cell (a, b) is not a wall;
- The sum of $n \times m$ of all test cases will not exceed 2×10^4.

Output

For each test case output one line containing one integer, indicating the number of pieces of litter the robot can pick up after executing k instructions.

Example

standard input

```
2
5 5
2 4 6
RUPIRPIUDDLRUDRURLIIURDLPRDLDIRLIDPPRRRLLULPRPUPPDPRIUIUDLULIRIDIRPUPPIRRLRLU >>
LUPLRIIRLPRLLRLDLLPDRUUDLDPRRPLLPPUIUUPPLUIILLDRIDILDRRUPLPPLPDLDPDDUPIPPUIIL >>
IPLUPLURRPIIDDPPIUPRPRIRDRPPIUIRDUUUPPPDIIRPURIUIUIPLRILLDPPPURPPRRPDPRRLUDUD >>
UDUPRLIUIRLR
11111
12021
10101
12021
11111
4 5
2 2 1000000000000000000
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII >>
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII >>
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII >>
IIIIIIIIIIII
11111
10021
11211
11111
```

standard output

```
2
0
```

Note

In the above examples, "»" indicates that the contents of the current line and the next line is actually on the same line in input data. We have to split them into different lines here because they're too long to fit into one line of the printed problemset. See the contest website for a more precise input example.

Problem D. Robot Cleaner II

(Please read the description of Robot Cleaner I first.)

Tired of programming the robot every time according to the situation of his room, BaoBao decides to write a "super program" so that the robot can pick up most of the litter using this program, no matter how the litter is positioned in the room.

To simplify this problem, we now put some constraints on BaoBao's room and the robot.

- $n = m = 12$;
- For all $1 < i < n$ and $1 < j < m$, (i, j) is NOT a wall;
- There are exactly 50 pieces of litter in the room, and they are randomly distributed with equal probability;
- The robot always starts from $(2, 2)$.

Your task is to design the "super program", so that when feeding this program to the controller, the robot can pick up at least 95% of the litter in the room under the above constraints after executing 200 instructions.

Input

There is no input for this problem.

Output

You should output a string of length 243 ($243 = 3^5$) consisting of 'U', 'D', 'L', 'R', 'P' and 'I' in one line, indicating your super program.

Example

standard input
(No input)

standard output
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII >> IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII >> IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII >> IIIIIIIIIIII

Note

In the above example, "»"indicates that the contents of the current line and the next line is actually on the same line in the output. We have to split them into different lines here because they're too long to fit into one line of the printed problemset. See the contest website for a more precise output example.

Also note that this output example is (obviously) NOT the correct answer. It only serves the purpose of showing you the output format.

We have prepared 1000 randomly generated rooms under the above constraints and will test your solution against them. Your solution will be accepted if the robot can pick up at least 95% of the litter **on average** in these 1000 rooms after executing 200 instructions in your super program.

Problem E. Potion

BaoBao is brewing a magical potion. To brew this potion, n types of ingredients, whose rank ranges from 1 to n, is needed. More precisely, for all $1 \le i \le n$, BaoBao needs at least a_i pieces of rank-i ingredients to make the potion, while he only has b_i pieces of these ingredients in his storeroom.

Fortunately, BaoBao is able to downgrade a higher rank ingredient to a lower rank one (this operation can be performed any number of times, including zero time). Is it possible that BaoBao can make the potion using the ingredients in his storeroom?

Input

There are multiple test cases. The first line of the input contains an integer T (about 100), indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 100$), indicating the number of types of ingredients.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$), where a_i indicates the number of rank-i ingredients needed.

The third line contains n integers $b_1, b_2, \dots, b_n$ ($1 \le b_i \le 10^9$), where b_i indicates the number of rank-i ingredients BaoBao has in his storeroom.

Output

For each test case output one line. If BaoBao is able to brew the potion, output "Yes" (without quotes), otherwise output "No" (without quotes).

Example

standard input	standard output
2 3 3 3 1 1 2 5 3 3 1 2 5 2 1	Yes No

Note

For the first sample test case, BaoBao can downgrade one rank-3 ingredient to a rank-2 ingredient, and downgrade two rank-3 ingredients to two rank-1 ingredients.

Problem F. Strange Function

If a string s is composed of digits from 0 to 9, we call this string a "digit string". It's obvious that each substring of s, except those who start with 0, can be considered as a positive integer. Let $\mathbb{S}$ be the set containing all these positive integers, define a function

$$f(s) = \min\{x | x \in \mathbb{Z}^+, x \notin \mathbb{S}\}$$

That is to say, $f(s)$ is the smallest positive integer which is not in $\mathbb{S}$. For example, $f(\text{"1241"}) = 3$, $f(\text{"542"}) = 1$, $f(\text{"11023456879"}) = 12$, and $f(\text{"00000"}) = 1$ (note that in this case, $\mathbb{S}$ is an empty set).

Given a digit string s, let $\text{substr}(i, j)$ be the substring of s which starts from the i-th digit and ends at the j-th digit, your task is to calculate

$$\sum_{i=1}^{|s|} \sum_{j=i}^{|s|} f(\text{substr}(i, j))$$

.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first and only line contains a digit string s ($1 \le |s| \le 5 \times 10^5$).

It's guaranteed that the sum of $|s|$ of all test cases will not exceed 5×10^6.

Output

For each test case output one line containing one integer, indicating the answer.

Example

standard input	standard output
4 1241 542 11023456789 00000	21 6 162 15

Note

We explain the first sample test case below.

Substring	Value	Substring	Value
1	2	24	1
2	1	41	2
4	1	124	3
1	2	241	3
12	3	1241	3

So the answer is $2 + 1 + 1 + 2 + 3 + 1 + 2 + 3 + 3 + 3 = 21$.

For the second sample test case, it's obvious that the values of all the substrings are 1. So the answer is 6.

Problem G. Postman

N letters have just arrived at the post office positioned at $x = 0$, and the i-th letter should be posted to position $x = a_i$. BaoBao, our beloved postman, will start his work from the post office and deliver all these letters by himself.

Unfortunately, BaoBao's backpack can only hold at most K letters each time (which means that if he wants to deliver some letter not in his backpack, he will have to go back to the post office and fetch it), so he may not be able to deliver all N letters in one go. Please note that BaoBao cannot temporarily drop a letter outside the post office and pick it back afterward.

What's the minimum distance BaoBao has to travel to deliver all N letters?

It's NOT necessary that BaoBao ends his delivery in the post office.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers N and K ($1 \le K \le N \le 10^5$), indicating the total number of letters and the capacity of the backpack.

The second line contains N integers $a_1, a_2, \ldots, a_N$ ($-10^9 \le a_i \le 10^9$), indicating the destination of each letter.

It's guaranteed that the sum of N over all test cases will not exceed 10^6.

Output

For each test case output one line containing one integer, indicating the minimum distance BaoBao has to travel to deliver all the letters, starting from the post office at $x = 0$.

Example

standard input	standard output
2 5 3 -1 -2 3 -4 -5 6 3 1 0 -2 -1 1 2	13 6

Note

For the first sample test case, BaoBao can first deliver the 1st and the 3rd letter (go to $x = 3$, then to $x = -1$, then to the post office), then deliver the 2nd, the 4th and the 5th letter (go to $x = -2$, then to $x = -4$, then to $x = -5$), and ends his delivery at $x = -5$.

Problem H. Rescue the Princess

Princess Cjb is caught by Heltion again! Her knights Little Sub and Little Potato are going to Heltion Kingdom to rescue her.

Heltion Kingdom is composed of n islands, numbered from 1 to n. There are m bridges in the kingdom, among which the i-th bridge connects the l_i-th island and the r_i-th island. The knights can go through each bridge in both directions.

Landing separately on the v-th and the w-th island, the two knights start their journey heading to the u-th island where the princess is imprisoned. However, as the knights are fat and the bridges are unstable, there will be a risk of breaking down the bridge and falling into the water if they go through one or more common bridges during their journey.

Thus, to successfully bring back the princess, two paths **with no common bridges** are needed: one starts from the v-th island and leads to the u-th island, while the other starts from the w-th island and also leads to the u-th island.

As the princess is caught very often, the knights will ask you for help q times. Each time, given their starting islands and their goal, you need to tell them whether it's possible to find two paths satisfying the constraints above.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains three integers n, m and q ($1 \le n \le 10^5$, $0 \le m \le 2 \times 10^5$, $1 \le q \le 10^5$), indicating the number of islands, the number of bridges and the number of queries.

The following m lines describe the bridges. The i-th line contains two integers l_i and r_i ($1 \le l_i, r_i \le n$), indicating the two islands the i-th bridge connects. Notice that different bridges may connect the same pair of islands and a bridge may connect an island to itself.

The following q lines describe the queries. The i-th line contains three integers u_i, v_i and w_i ($1 \le u_i, v_i, w_i \le n$), indicating the island where the princess is imprisoned and the starting islands of the two knights.

It's guaranteed that the sum of n of all test cases will not exceed 5×10^5, the sum of m of all test cases will not exceed 10^6, and the sum of q of all test cases will not exceed 5×10^5.

Output

For each test case output q lines indicating the answers of the queries. For each query, if two paths meeting the constraints can be found, output "Yes" (without quotes), otherwise output "No" (without quotes).

Example

standard input	standard output
2 6 7 4 1 2 2 3 3 1 4 5 5 6 6 4 1 4 4 1 3 1 4 2 1 2 3 1 3 3 2 1 2 1 2 1 1 1 2 1 2	No Yes Yes Yes Yes Yes

Note

For the first sample test case:

- For the 2nd query, we can select the paths 4-1 and 2-1.
- For the 3rd query, we can select the paths 2-1 and 3-1.
- For the 4th query, we can select the paths 3-1 and 3-2-1.

For the second sample test case:

- For the 1st query, as the knights and the princess are on the same island initially, the answer is "Yes".
- For the 2nd query, as one of the knights are on the same island with the princess initially, he does not need to cross any bridge. The other knight can go from island 1 to island 2 directly.

Problem I. Defense Plan

Potato Kingdom is about to be attacked by the evil Heltion Kingdom! As the Secretary of Defense, Little Sub must arrange the defense system very carefully.

There are n castles in Potato Kingdom in total and Little Sub is planning to equip some of them with cannons. However, if two castles are too close to each other, they cannot be both cannon-armed since they might hurt each other.

The i-th castle has a defense value of w_i. Let $\mathbb{C}$ be the set of the **index** of all castles armed with cannons, we define the total defense value to be

$$\prod_{k \in \mathbb{C}} w_k$$

For completeness, if $\mathbb{C} = \emptyset$, the total defense value is defined to be 1.

There are many possible plans to equip cannons to these castles and Little Sub must choose a proper one. In order to value the situation correctly and more precisely, he wants to know about the variance S^2 of all feasible plan's total defense value (note that arming no castle with cannons is also a feasible plan). Two plans A and B are considered to be different if a castle is armed with cannons in plan A but not in plan B, or vice versa.

Recall that the variance S^2 of k values $x_1, x_2, \ldots, x_k$ can be calculated as follows:

$$\bar{x} = \frac{x_1 + x_2 + ... + x_k}{k} \qquad S^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + ... + (x_k - \bar{x})^2}{k}$$

Input

There will be only one test case in each test file (and of course we have prepared multiple test files).

The first line contains two integers n and m ($1 \le n \le 40, 0 \le m \le \frac{n(n-1)}{2}$), indicating the number of castles and conflict relations.

The second line contains n integers $w_1, w_2, \ldots, w_n$ ($0 \le w_i \le 2^{31} - 1$), indicating the defense value of castles.

For the following m lines, the i-th line contains two integers x_i and y_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$), indicating that the x_i-th and the y_i-th castle cannot be both armed with cannons.

Output

Output a single integer in a single line, indicating the answer. If the answer is $\frac{A}{B}$, please print $C (0 \le C < 10^9 + 7)$ where $A \equiv C \times B \pmod{10^9 + 7}$.

Example

standard input	standard output
3 1 1 2 3 1 2	888888898

Note

Castles Armed	Defense Value	Castles Armed	Defense Value
{}	1	{3}	3
{1}	1	{1, 3}	3
{2}	2	{2, 3}	6

So the answer is $\frac{26}{9}$. As $9 \times 888888898 = 8000000082 \equiv 26 \pmod{10^9 + 7}$, we print 888888898.

Problem J. Extended Twin Composite Number

Do you know the twin prime conjecture? Two primes a and b are called twin primes if $a + 2 = b$. The twin prime conjecture is an unsolved problem in mathematics, which asks for a proof or a disproof for the statement "there are infinitely many twin primes".

On April 17, 2013, Yitang Zhang announced a proof that for some integer c that is less than 70 million, there are infinitely many pairs of primes that differ by c. As of April 14, 2014, one year after Zhang's announcement, the bound has been reduced to 246. People are hoping for the bound to be smaller and smaller, so that a proof for the conjecture can finally be found.

For our dear contestants, we've prepared another similar problem for you, which is the extended twin composite number problem: Given a positive integer n, find two integers x and y such that $x + n = y$ and both x and y are composite numbers.

Input

There are multiple test cases. The first line of the input contains an integer T (about 10^5), indicating the number of test cases. For each test case:

The only line contains one integer n ($1 \le n \le 10^9$).

Output

For each test case output two integers in one line, indicating x and y where $1 \le x, y \le 10^{18}$. If there are multiple valid answers, you can print any of them; If there is no valid answer, output "-1" (without quotes) instead.

Example

standard input	standard output
3 11 1805296 5567765	4 15 114514 1919810 111234 5678999