1001 Static Query on Tree

Time Limit: 2000/1000 MS (Java/Others)

Memory Limit: 262144/262144 K (Java/Others)

Problem Description

In country X, there are n cities and $n-1$ **one-way roads**, and all city can reach city 1. One query will give 3 sets of cities A, B, C. Alice will choose a city x in set A, choose a city z in set C, and walk from x to z (if x can reach z). Bob will choose a city y in set B, and walk from y to z (if y can reach z). How many cities can possibly be the city where Alice and Bob meet each other?

In other words, how many cities can be reached from any city in set A, any city in set B, and can reach any city in set C?

There are T test cases, and each case has q queries.

Input

First line is one integer T, indicating T test cases. In each case:

First line is 2 integers n, q, indicating n cities and q queries.

Next line is $n-1$ integers $r_1, r_2, \dots, r_{n-1}$, the i-th integer indicates the road from city $i+1$ to city r_i.

Next is q queries, in each query:

First line is 3 integer $|A|, |B|, |C|$, indicating the size of set A, B, C.

Next line is $|A|$ integers, indicating the set A.

Next line is $|B|$ integers, indicating the set B.

Next line is $|C|$ integers, indicating the set C.

$1 \leq T \leq 20, 1 \leq n, q, |A|, |B|, |C| \leq 2 \times 10^5$, for all cases $\sum n \leq 2 \times 10^5$, $\sum q \leq 2 \times 10^5$, for all queries in all cases $\sum |A| + \sum |B| + \sum |C| \leq 2 \times 10^5$.

Output

In each case, print q integers, one integer per line, i-th integer indicates the answer of i-th query.

Sample Input

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

Sample Output

```
2
4
1
```

1002 C++ to Python

Time Limit: 2000/1000 MS (Java/Others)

Memory Limit: 262144/262144 K (Java/Others)

Problem Description

Kayzin's repository has lots of code in C++, but Aba aba can only understand Python code. So Aba aba calls you for help.

The only things you should do is using (...) in Python to match std::make_tuple(...) in C++.

Input

First line has one integer T, indicates T test cases. In each case:

First line has string s, indicates C++ code.

The C++ code only contains std::make_tuple, "(", ")", "," and integers, without space.

$1 \leq T \leq 100$, the length of s not exceeds 1000.

Output

Each test cases print one line of Python code, do not print any space.

Sample Input

```
2
std::make_tuple(-2,3,3,std::make_tuple(3,3))
std::make_tuple(std::make_tuple(1,1,4,5,1,4))
```

Sample Output

```
(-2,3,3,(3,3))
((1,1,4,5,1,4))
```

1003 Copy

Time Limit: 4000/2000 MS (Java/Others)

Memory Limit: 262144/262144 K (Java/Others)

Problem Description

Kayzin has a list of integers, initially the list is $a_1, a_2, \ldots, a_n$. He will execute q operations.

For an operation of first type, he will select an interval $[l_i, r_i]$, copy and insert it to the end of the interval.

For an operation of second type, he wonder the x_i-th integer of the list.

You need to print the xor sum of all the answers of second type operations.

ps: What is xor? The xor value of two integers is equal to addition in binary without carry.

Input

First line is an integer T, indicating the number of test cases. For each test case:

First line is 2 integers n, q, indicating the length of initial list and the number of operations.

Next line is n integers $a_1, a_2, \ldots, a_n$, indicating the initial list.

Next q line, one operation per line. The i-th line could be 3 integers $(1, l_i, r_i)$, indicating the first type operation, or 2 integers $(2, x_i)$, indicating the second type operation.

$1 \leq T \leq 10, 1 \leq n, q \leq 10^5, 1 \leq a_i \leq 10^9, \sum n \leq 10^5, \sum q \leq 10^5, 1 \leq x_i, l_i, r_i \leq n$, the number of first type operations not exceeds 20000.

Output

For each test case, print one line, indicating the xor sum of the answers.

Sample Input

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

Sample Output

```
6
```

Hint

For first operation, the 4-th integer is 4.
For second operation, $2, 3, 4$ is copied, the list becomes $1, 2, 3, 4, 2, 3, 4, 5$.
For third operation, the 5-th integer is 2.
So the result is 2 xor 4 = 6

1004 Keychains

Time Limit: 2000/1000 MS (Java/Others)

Memory Limit: 262144/262144 K (Java/Others)

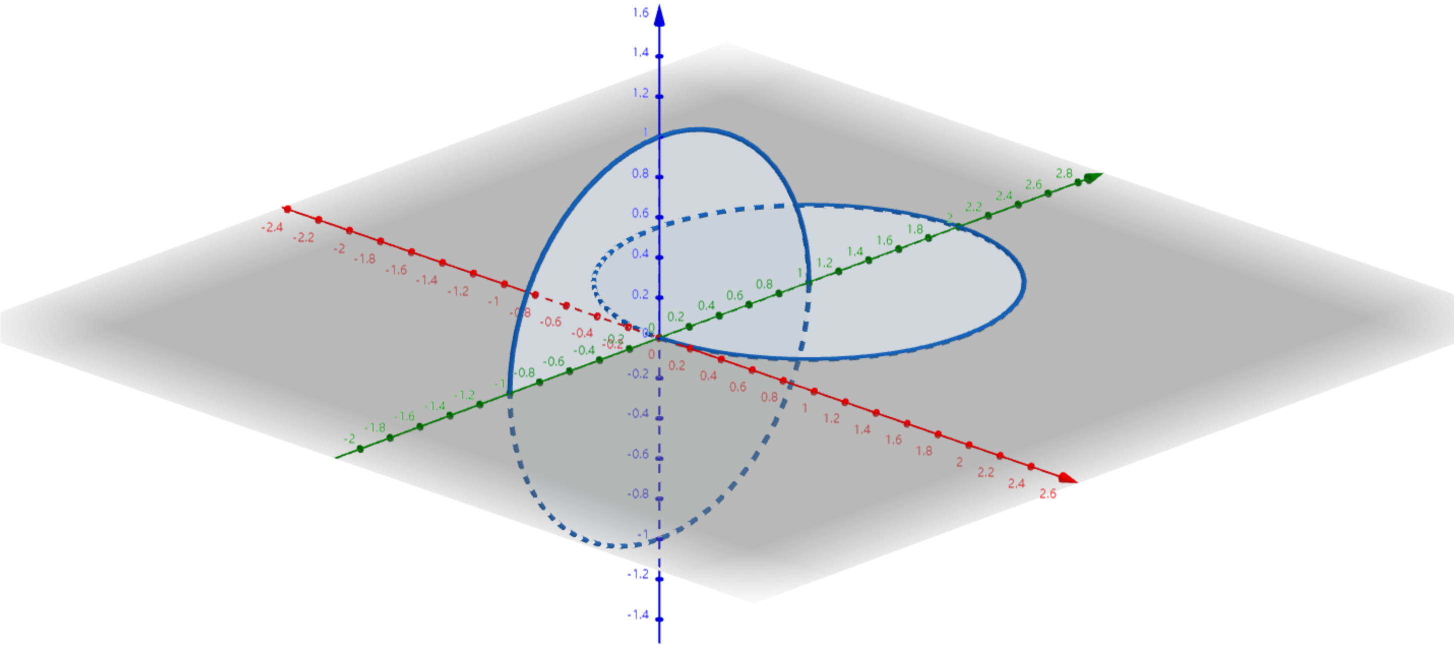
Problem Description

Kayzin is a magician. During a performance one day, he took out two keychains and covered them with a cup. He claimed that when the cup were lifted, the two keychains would be interlocked. But because of the poor lighting, the audience can't see clearly, so the audience wants you to figure out whether the two keychains are interlocked. If yes, please print "Yes", otherwise print "No".

A keychain can be described as a circle in 3-dimensional space.

Two keychains are interlocked, if and only if we can't let their center's distance greater than 10^{114514} by moving two keychains without collision.

For the first sample, two keychains are as follows, they are interlocked obviously.



Input

First line is one integer T, indicating T test cases. In each case:

First line is 7 integers $x_1, y_1, z_1, x_2, y_2, z_2, r$, (x_1, y_1, z_1) indicates the coordinate of the center of the first circle, (x_2, y_2, z_2) indicates the normal vector of the plane of the first circle, r indicates the radius of the first circle.

Second line is 7 integers $x_1', y_1', z_1', x_2', y_2', z_2', r'$, (x_1', y_1', z_1') indicates the coordinate of the center of the second circle, (x_2', y_2', z_2') indicates the normal vector of the plane of the second circle, r' indicates the radius of the second circle.

It is guaranteed that the distance between any two points on two circles is not less than 0.1.

$$1 \le T \le 1000, -1000 \le x_1, y_1, z_1, x_2, y_2, z_2, x_1', y_1', z_1', x_2', y_2', z_2' \le 1000, 1 \le r, r' \le 1000$$

Output

For each test case, print one line, "Yes" or "No", indicating answer.

Sample Input

```
2
0 0 0 1 0 0 1
0 1 0 0 0 1 1
0 0 0 1 0 0 1
0 1 0 0 0 1 10
```

Sample Output

```
Yes
No
```

1005 Slayers Come

Time Limit: 3000/1500 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

Problem Description

Kayzin has recently become addicted to a game called Slayers Come. The game opens with n monsters standing in a line, with the i-th monster having an attack power of a_i (the amount of damage the monster deals when it launches an attack) and a defense power of b_i (the amount of damage the monster can mitigate when it takes an attack).

Kayzin has m skills to learn, with the i-th skill allowing Kayzin to directly defeat a monster with subscript x_i. This skill has a death rattle effect, i.e., if $monster_{x_i}$ is defeated and there is a monster to its left (subscripted $x_i - 1$), $monster_{x_i}$ will launch an attack with damage $a_{x_i} - L_i$ against $monster_{x_i - 1}$; if there is a monster to its right (subscripted $x_i + 1$), then $monster_{x_i}$ also fires an attack with damage $a_{x_i} - R_i$ at $monster_{x_i+1}$).

If the damage dealt (Damage value - current monster defense) to the monster by one attack is greater than or equal to 0, the monster is defeated , conversely the attack is invalid. It should be noted that when a monster dies, the death rattle causes a chain reaction, meaning that the monster defeated by the death rattle will then attack the monsters on either side of it.Namely,

- When Kayzin defeats $monster_j$ with the i-th skill (by direct attack or deathrattle), if $j > 1$ and $a_j - L_i \geq b_{j-1}$, then this skill also defeats $monster_{j-1}$
- When Kayzin defeats $monster_j$ with the i-th skill (by direct attack or deathrattle), if $j < n$ and $a_j - R_i \geq b_{j+1}$, then this skill also defeats $monster_{j+1}$

All monsters, including the defeated monsters, always keep their subscripts constant. The defeated monster will re-generate after the effects of all attacks caused by the current skill end, and the re-generated monster keeps its original attack and defense power unchanged.

Kayzin would like to know how many options for learning skills that make it possible to defeat every monster **at least once** after releasing all the learned skills. The answer modulo 998244353.

Input

The first line contains an integer $T(T \leq 100)$. Then T test cases follow. For one case,

The first line contains two integer $n\ (n \leq 10^5)$ and $m\ (m \leq 10^5)$, n denotes the total number of monsters, and the subscripts of monsters from left to right are 1~n. m denotes the type of skills that kayzin can learn.

The next n lines lists the attack and defense power of the monsters. The i-th line has two numbers, a_i and $b_i\ (1 \leq a_i, b_i \leq 10^9)$, a_i denotes the attack power of the $monster_i$, b_i denotes the defense power of the $monster_i$.

The next m lines lists the target of the skill's attack and the effects of its death rattle. The i-th line has three numbers, $x_i\ (1 \leq x_i \leq n)$, L_i and $R_i\ (-10^9 \leq L_i, R_i \leq 10^9)$, x_i denotes the attack target of the i-th skill, L_i denotes how much the monster defeated by this skill weakens the attack power of the monster to its left, R_i the same way.

It is guaranteed that the sum of n over all test cases doesn't exceed 10^5 and the sum of m over all test cases doesn't exceed 10^5.

Output

Print an integer for each case, indicating the number of options for learning skills that make it possible to defeat all the monsters at least once after releasing all the learned skills, the answer modulo 998244353.

Sample Input

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

Sample Output

1006 Bowcraft

Time Limit: 4000/2000 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

Problem Description

Kayzin has created a game called $Bowcraft$. In the game you will play as an archer and at the beginning of the game you will get a bow with a level of 0. To get a more friendly game experience, you can buy enhancement books to upgrade your bow.

Each enhancement book has two attribute. $\frac{a}{A}$ is the probability of successfully upgrading your bow by 1 level when you use the book; if the upgrade fails, the bow will have a $\frac{b}{B}$ probability of breaking(reduced to level 0). When you buy an enhancement book, the store system will generate a random integer in $[0, A-1]$ with equal probability as attribute a and a random integer in $[0, B-1]$ with equal probability as attribute b.

After you buy an enhancement book, you need to choose to use this enhancement book or discard it.

Kayzin would like to quiz the clever you on the best strategy to expectation of enhancement books you need to buy to upgrade your bow from level 0 to level K.

Input

The first line contains an integer $T(T \leq 10)$,denoting the number of groups of data. Next follows T lines, for each line, there has three integers K $(1 \leq K \leq 1000)$, A , B $(2 \leq A, B \leq 100)$, indicating that you need to raise the bow to level K, as well as the range of upgrade success rate A and the range of breakage rate B using each enhancement book.

Output

Output the expectation of enhancement books to be purchased to raise a bow from 0 to K level, and the result are reserved with 3 digits after the decimal point.

Sample Input

```
3
1 3 3
2 3 3
319 100 100
```

Sample Output

```
3.000
7.000
13436.938
```

1007 Snatch Groceries

Time Limit: 2000/1000 MS (Java/Others)

Memory Limit: 65536/65536 K (Java/Others)

Problem Description

“SNATCH GROCERIES first, then get a covid test” has quickly become an anthem for the lockdown that started suddenly in Shanghai in the early hours of March 28th. We can describe scenes of panic buying—qiang cai, or snatching groceries—and the threat of being locked out of one’s home amid a frenzied bid to control an outbreak of covid-19 in China’s main business and finance hub. Here is the question, how does the server determine who succeeded When millions of people press the order button on their mobile phones at the same time.

Processing such a large number of requests requires a distributed system. In a distributed system, time is a tricky business, because communication is not instantaneous: it takes time for a message to travel across the network from one machine to another. The time when a message is received is always later than the time when it is sent, but due to variable delays in the network, we don’t know how much later. This fact sometimes makes it difficult to determine the order in which things happened when multiple machines are involved

Moreover, each machine on the network has its own clock, which is an actual hardware device: usually a quartz crystal oscillator. These devices are not perfectly accurate, so each machine has its own notion of time, which may be slightly faster or slower than on other machines. It is possible to synchronize clocks to some degree: the most commonly used mechanism is the Network Time Protocol (NTP), which allows the computer clock to be adjusted according to the time reported by a group of servers. The servers in turn get their time from a more accurate time source, such as a GPS receiver.

Modern computers have at least two different kinds of clocks: a time-of-day clock and a monotonic clock. A time-of-day clock does what you intuitively expect of a clock: it returns the current date and time according to some calendar (also known as wall-clock time). For example, clock_gettime(CLOCK_REALTIME) on Linux return the number of seconds (or milliseconds) since the epoch: midnight UTC on January 1, 1970, according to the Gregorian calendar, not counting leap seconds. Some systems use other dates as their reference point. Time-of-day clocks are usually synchronized with NTP, You may be able to read a machine's time-of-day clock with microsecond or even nanosecond resolution. But even if you can get such a fine-grained measurement, that doesn't mean the value is actually accurate to such precision. In fact, it most likely is not—as mentioned previously, the drift in an imprecise quartz clock can easily be several milliseconds, even if you synchronize with an NTP server on the local network every minute. With an NTP server on the public internet, the best possible accuracy is probably to the tens of milliseconds, and the error may easily spike to over 100 ms when there is network congestion.

Thus, it doesn't make sense to think of a clock reading as a point in time—it is more like a range of times, within a confidence interval: for example, a system may be 95% confident that the time now is between 10.3 and 10.5 seconds past the minute, but it doesn't know any more precisely than that. If we only know the time +/– 100 ms, the microsecond digits in the timestamp are essentially meaningless.

An interesting exception is Google's TrueTime API in Spanner, which explicitly reports the confidence interval on the local clock. When you ask it for the current time, you get back two values: [earliest, latest], which are the earliest possible and the latest possible timestamp. Based on its uncertainty calculations, the clock knows that the actual current time is somewhere within that interval. The width of the interval depends, among other things, on how long it has been since the local quartz clock was last synchronized with a more accurate clock source.

TL;DR: Spanner implements snapshot isolation across data centers in this way. It uses the clock's confidence interval as reported by the TrueTime API, and is based on the following observation: if you have two confidence intervals, each consisting of an earliest and latest possible timestamp ($A = [A_{earliest}, A_{latest}], B = [B_{earliest}, B_{latest}]$), and those two intervals do not overlap (i.e.,$A_{earliest} < A_{latest} < B_{earliest} < B_{latest}$), then B definitely happened after A——there can be no doubt. Only if the intervals overlap are we unsure in which order A and B happened.

Now we use Spanner as a solution, there are millions of people snatching groceries, and everyone is given the clock's confidence interval. The server executes each request in chronological order, and terminate in case of intervals overlap. Here is the question, how many people can get their food before the server is terminated.

Input

First line has one integer T, indicating there are T test cases. In each case:

First line has one integers n, indicating there are n people.

For next n lines, each line has 2 integers $earliest, latest$, indicates the clock's confidence interval.

$T \le 10, 1 \le n \le 10^5, 0 \le earliest_i < latest_i \le 10^9$

Output

In each case, print one integer, indicates the answer.

Sample Input

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

Sample Output

```
3
0
```

1008 Keyboard Warrior

Time Limit: 4000/2000 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

Problem Description

Some contestants said on the Internet that they love Multi-University Training, did the rest of them have no keyboards?

You must be the one whose keyboard is badly broken. When you press a key, it triggers a random number of times.

Given a character ch and an integer k, it means you press an alphanumeric key ch only once, but it triggers k times, and k character ch will be added to the end of the buffer.

Given a character `-` and an integer k, it means you press the backspace key, it triggers k times, delete k characters from the end (If the number of characters is less than k, the buffer will be cleared).

Given the operations in chronological order, could you input your target text? Which means whether there is a time, your target text is a substring of your buffer characters? Answer 'yes' or 'no'. (In formal language theory and computer science, a substring is a contiguous sequence of characters within a string.)

Input

First line has one integer T, indicating there are T test cases. In each case:

First line has two integers n, m, n indicates the length of your target text, m indicates the number of times you press the key.

Second line has a string of length n, which contains only lowercase letters.

For next m lines, each line has a character ch and an integer k, their meanings are described above.

$1 \le n, m \le 2 \times 10^5, 0 \le k \le 10^9, \sum n + m \le 10^6$

Output

In each case, print 'yes' or 'no', without quote.

Sample Input

```
3
6 6
iloveu
i 1
l 1
o 1
v 1
e 1
u 0
6 10
imfive
u 10
- 20
i 1
m 1
f 1
i 1
v 5
- 4
e 2
- 2
4 4
abab
a 2
b 2
- 3
b 1
```

Sample Output

```
no
yes
no
```

1009 ShuanQ

Time Limit: 2000/1000 MS (Java/Others)

Memory Limit: 65536/65536 K (Java/Others)

Problem Description

CX is a programmer of a mooc company. A few days ago, he took the blame for leakage of users' data. As a result, he has to develop an encryption algorithm, here is his genius idea.

First, the protocol specifies a prime modulus M, then the server generates a private key P, and sends the client a public key Q. Here $Q = P^{-1}, P \times Q \equiv 1 \mod M$.

Encryption formula: $encrypted\_data = raw\_data \times P \mod M$

Decryption formula: $raw\_data = encrypted\_data \times Q \mod M$

It do make sense, however, as a master of number theory, you are going to decrypt it.You have intercepted information about $P, Q, encrypted\_data$, and M keeps unknown. If you can decrypt it, output raw_data, else, say "shuanQ" to CX.

Input

First line has one integer $T (T \leq 20)$, indicating there are T test cases. In each case:

One line has three integers $P, Q, encrypted\_data$. $(1 < P, Q, encrypted\_data \leq 2 \times 10^6)$

It's guaranteed that $P, Q, encrypted\_data < M$.

Output

In each case, print an integer raw_data, or a string "shuanQ".

Sample Input

```
2
5 5 5
6 6 6
```

Sample Output

```
shuanQ
1
```

1010 Assassination

Time Limit: 4000/2000 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

Problem Description

Ampere (symbol: A), is the SI base unit of electric current. On a circuit board, there is a electric current called Ambere. He is different from any other currents because the wires he passed never complete the circuit (he can go back, but all wires he visited won't form the loop).

There are n nodes and m wires (wires are bi-directional) on the circuit board. Each wire connects two nodes, and i'th wire has w_i components. If a current passes a wire, the components on the wire would be able to work.

Ambere can start from any node, and stop at any node. Components on wires which Ambere visited once or more will be able to work. Ambere is a good-hearted current so he will pass as much components as he can.

But you're not a nice guy. As a member of IEC(International Electro technical Commission), you feel extremely angry that a current never complete the circuit. So you plan to send several killers to wires. If Ambere passes a wire with a killer, he will be assassinated.

Your commission offers limited funds, so you want to know what's the minimal number of killers you need?

Input

First line has one integer T, indicating there are T test cases. In each case:

First line has two integers n, m, indicating the number of nodes and the number of wires.

For next m lines, each line has 3 integers u, v, w, indicating a wire connecting node u and node v, and there're w components on the wire.

It is guaranteed that the graph is connected and doesn't contain loops or multiple edges, the number of wires with a same w would be no more than 10^2 .

$1 \le n, w_i \le 10^5, n-1 \le m \le \min(\frac{n(n-1)}{2}, 2 \times 10^5), \sum m \le 10^6$

Output

In each case, print one integer, indicates the answer.

Sample Input

```
2

3 3
1 2 2
1 3 1
2 3 1

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

Sample Output

```
1
3
```

1011 DOS Card

Time Limit: 4000/2000 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

Problem Description

DOS is a new single-player game that Kayzin came up with. At the beginning of the game you will be given n cards in a row, each with the number of value a_i.

In each "matching" operation you can choose any two cards (we assume that the subscripts of these two cards are $i, j(i < j)$.**Notice that i is less than j**), and you will get a score of $(a_i + a_j) \times (a_i - a_j)$.

Kayzin will ask you m times. In the k-th query, you need to select four cards from the cards with subscripts L_k to R_k, and "match" these four cards into two pairs (i.e., two matching operations, but the subscripts of the cards selected in the two matching operations must be different from each other. That is, a card can only be matched at most once. e.g., if you select four tickets with subscripts a, b, c, and d, matching a with b and c with d, or matching a with c and b with d are legal, but matching a with b and b with c is not legal), please calculate the maximum value of the sum of the two matching scores.

Note that the queries are independent of each other.

Input

The first line contains an integer $T(T \le 100)$. Then T test cases follow. For one case,

The first line contains two integer n $(4 \le n \le 2 \times 10^5)$ and m $(1 \le m \le 10^5)$, n denotes the total number of cards , m denotes the number of times Kayzin queries.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^8)$, denotes the value of each card.

The next m lines contain Kayzin's queries. The kth line has two numbers, L_k and R_k $(1 \le L_k \le R_k \le n)$, the input guarantees that $R_k - L_k \ge 3$.

It is guaranteed that the sum of n over all test cases doesn't exceed 2×10^5 and the sum of m over all test cases doesn't exceed 2×10^5.

Output

Print m integer for each case, indicating the maximum scores that can be obtained by selecting four cards (two matching pairs)

Sample Input

```
1
5 3
5 3 2 8 4
1 5
1 4
2 5
```

Sample Output

```
69
-34
53
```

1012 Luxury cruise ship

Time Limit: 2000/1000 MS (Java/Others)

Memory Limit: 524288/524288 K (Java/Others)

Problem Description

Kayzin plans to buy a luxury cruise ship which will cost N coins. But Kayzin didn't have enough coins, so Kayzin decided to start saving coins.

Kayzin will put 7 or 31 or 365 coins into his piggy bank each day. Now Kayzin wants to know at least how many days he will need to "exactly"(No more and no less) scrape together the money for buying a luxury cruise.

If Kayzin can't "exactly" scrape together the money for buying a luxury cruise, print -1.

Input

The first line contains an integer $T(T \leq 1000)$. Then T test cases follow.

For one case, the first line contains an integer N $(1 \leq N \leq 10^{18})$, N denotes the coins that a luxury cruise ship will cost.

Output

Print an integer for each case, indicating the minimum number of days Kayzin needs.

If Kayzin can't "exactly" scrape together the money for buying a luxury cruise, print -1.

Sample Input

```
5
14
38
55
403
996
```

Sample Output

```
2
2
-1
3
16
```