

А. Хитрый шифр

Ограничение времени	2 секунды
Ограничение памяти	512Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Известная компания Тындекс в очередной раз проводит набор стажёров.

Забываясь о персональных данных соискателей, компания придумала хитрый алгоритм шифрования:

- Подсчитывается количество различных символов в ФИО (регистр важен, А и а — разные символы).
- Берётся сумма цифр в дне и месяце рождения, умноженная на 64.
- Для первой (по позиции в слове) буквы фамилии определяется её номер в алфавите (в 1-индексации), умноженный на 256 (регистр буквы не важен).
- Полученные числа суммируются.
- Результат переводится в 16-чную систему счисления (в **верхнем регистре**).
- У результата сохраняются только 3 младших разряда (если значимых разрядов меньше, то шифр дополняется до 3-х разрядов ведущими нулями).

Ваша задача — помочь вычислить для каждого кандидата его шифр.

Формат ввода

В первой строке вводится число N ($1 \leq N \leq 10000$) — количество кандидатов и шифров.

Далее следует N строк в формате CSV ($f_j, i_j, o_j, d_j, m_j, y_j$) — информация о кандидатах:

- Фамилия f_j , имя i_j и отчество o_j ($1 \leq |f_j|, |i_j|, |o_j| \leq 15$) — строки, состоящие из латинских букв верхнего и нижнего регистра;
- день рождения d_j , месяц рождения m_j и год рождения y_j — целые числа, задающие **корректную** дату в промежутке от 1 января 1950 года до 31 декабря 2021 года.

Формат вывода

В единственной строке выведите N строк k_1, k_2, \dots, k_N , где k_j — шифр j -го кандидата (в **верхнем регистре**). Кандидаты нумеруются с 1 до N в порядке ввода.

Пример

Ввод

2

Volozh, Arcady, Yurievich, 11, 2, 1964

Segalovich, Ilya, Valentinovich, 13, 9, 1964

Вывод

710 64F

Примечания

Рассмотрим тестовый пример.

Первый кандидат — Volozh, Arcady, Yurievich, 11, 2, 1964:

- Различные символы в ФИО: V, o, l, z, h, A, r, c, a, d, y, Y, u, i, e, v - всего их **16**.
- Сумма цифр в дне и месяце рождения равна $1+1+2=4$.
- Номер в алфавите первой буквы фамилии V равен **22**.
- Итоговое значение шифра равно $16+4 \cdot 64+22 \cdot 256=5904$.
- В 16-ричной системе счисления это число представимо как **1710**.
- Нас интересуют только 3 последние разряда, поэтому остаётся **710**.

Второй кандидат — Segalovich, Ilya, Valentinovich, 13, 9, 1964:

- Различные символы в ФИО: S, e, g, a, l, o, v, i, c, h, I, y, V, n, t - всего их **15**.
- Сумма цифр в дне и месяце рождения равна $1+3+9=13$.
- Номер в алфавите первой буквы фамилии S равен **19**.
- Итоговое значение шифра равно $15+13 \cdot 64+19 \cdot 256=5711$.
- В 16-ричной системе счисления это число представимо как **164F**.
- Нас интересуют только 3 последние разряда, поэтому остаётся **64F**.

В. Через тернии к клиенту

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	1 секунда	512Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Python 3.7.3	4 секунды	512Mb		
Python 3.7 (PyPy 7.3.3)	4 секунды	512Mb		
Scala 2.13.4	5 секунд	512Mb		
PHP 7.3.5	4 секунды	512Mb		
OpenJDK Java 15	5 секунд	512Mb		
Kotlin 1.5.32 (JRE 11)	5 секунд	512Mb		

Известная компания Тындекс идёт в ногу со временем — с началом активных космических перелётов в компании открылся сервис Тындекс.Ракетакси: пользователю необходимо лишь указать координаты начала и конца перелёта, после чего за ним вылетит персональная ракета.

По сути любой заказ можно описать в виде событий четырёх типов:

1. **A** (accepted) - заказ принят в работу (ракета вылетела за клиентом);
2. **B** (boarding) - клиент сел в ракету;
3. **S** (success) - заказ успешно завершён (клиент вышел на планете назначения);
4. **C** (cancelled) - заказ отменён.

Все происходящие с ракетами события отправляются на сервера, где сразу логируются. Вот только из-за проблем со связью (метеоритные потоки, вспышки на звездах и т.д.) отправка событий иногда затягивается, из-за чего записи в получившемся логе могут идти не по порядку.

Гарантируется, что все описанные в логе события задают один из следующих сценариев:

1. **A - B - S**
2. **A - B - C**
3. **A - C**

Вам, как главному аналитику (и по совместительству главному программисту) ракетопарка, необходимо исследовать лог за прошедший год и определить для каждой ракеты суммарное время движения (в минутах) в течение заказов.

В каждый момент времени ракета выполняет только один заказ. Будем считать, что каждая ракета в каждый момент времени:

- или стоит на месте в ожидании заказа,
- или перемещается по космосу, выполняя заказ.

Движение начинается после принятия заказа и завершается после отмены или завершения заказа. За одну минуту не может произойти несколько событий, связанных с одной и той же ракетой.

Формат ввода

В первой строке дано целое число $N (2 \leq N \leq 200000)$ — количество записей в логе.

Следующие N строк содержат записи в логе в формате dayhourminuteidstatus:

- day($1 \leq d \leq 365$) — номер дня (сквозная нумерация с начала календарного года);
- hour($0 \leq h < 24$) — часы;
- minute($0 \leq m < 60$) — минуты;
- id($0 \leq id \leq 10^9$) — уникальный идентификатор ракеты;
- status $\in \{A, B, S, C\}$ — буква, обозначающая тип события.

Формат вывода

В единственной строке выведите через пробел суммарное время движения на заказах для каждой упомянутой в логе ракеты. Данные необходимо выводить в порядке возрастания идентификаторов ракет.

Пример

Ввод

Вывод

```
8
50 7 25 3632 A
14 23 52 212372 S
15 0 5 3632 C
14 21 30 212372 A 156 142
50 7 26 3632 C
14 21 30 3632 A
14 21 40 212372 B
14 23 52 3632 B
```

Примечания

Ракета №3632

1. в 14-й день года в 21:30 получила заказ (шестая запись в логе);
2. забрала пассажира в 23:52 того же дня (восьмая запись в логе);
3. после чего заказ был отменён в 15-й день года в 00:05 (третья запись в логе);
4. в 50-й день года в 7:25 получила заказ (первая запись в логе);
5. заказ был отменён уже через минуту (четвёртая запись в логе).

Таким образом ракета №3632 провела в движении с 14-го дня 21:30 до 15-го дня 00:05 и с 50-го дня 7:25 до 50-го дня 7:26 — всего 156 минут.

Ракета №212372

1. в 14-й день года в 21:30 получила заказ (третья запись в логе);
2. через 10 минут забрала пассажира (седьмая запись в логе);
3. в 23:52 прибыла на место назначения (вторая запись в логе).

Всего ракета №212372 провела в движении с 14-го дня 21:30 до 14-го дня 23:52 — 142 минуты.

С. Приснится же такое...

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	2 секунды	512Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Python 3.7.3	3 секунды	512Mb		
Python 3.7 (PyPy 7.3.3)	3 секунды	512Mb		
Scala 2.13.4	3 секунды	512Mb		
OpenJDK Java 15	3 секунды	512Mb		
PHP 7.3.5	3 секунды	512Mb		
Kotlin 1.5.32 (JRE 11)	3 секунды	512Mb		

Наконец-то с царством Морфея удалось наладить дипломатические отношения! Первым делом в магазины поступили самые корректные и полные сонники, составленные в сотрудничестве с главными сномагами царства.

Ваш близкий друг Тирания Вампадур купила такой сонник одной из первых. Но тут же её ждало разочарование. Оказалось, что некоторые сны образуют целую последовательность сюжетов, которую надо интерпретировать только целиком.

И у Тирании оказалась именно такая ситуация. Когда-то давно ей приснилось двоичное дерево из N вершин, занумерованных целыми числами от 1 до N .

Вершина 1 являлась корнем. У каждой вершины v было до двух сыновей: левый имел номер $2 \cdot v$, правый — $2 \cdot v + 1$ (при условии, что их номера не превосходили N). Таким образом, зная число N , дерево можно было однозначно восстановить.

Но, к сожалению, следующие Q ночей Тирании снились похожие сны: одна из вершин дерева v менялась местами с её предком (если v была корнем дерева, то ничего не происходило). Причем эти изменения переносились между снами, всё больше и больше изменяя оригинальное дерево.

Чтобы верно интерпретировать значение снов, Тирании нужно узнать итоговую структуру дерева после всех произошедших с ним изменений. Она просит вас помочь ей и по последовательности менявшихся вершин найти итоговую структуру дерева из её снов.

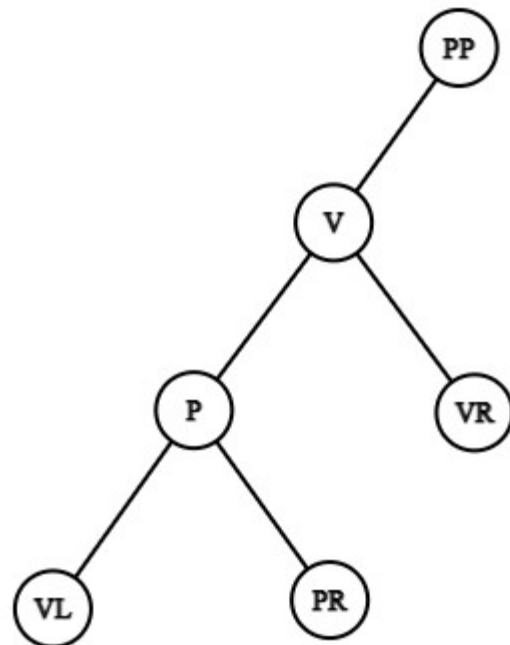
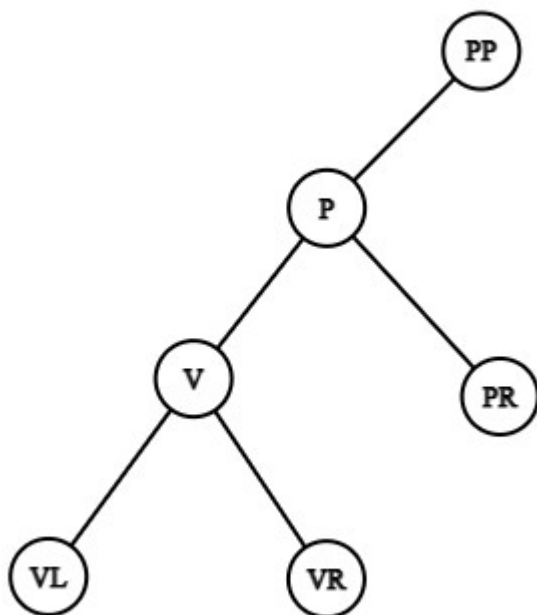
Понимая, что в этом деле важна точность, вы расспросили Тиранию насчет процесса обмена местами вершины v с её предком.

Введем обозначения:

- p — предок вершины v , pp — предок вершины p (если таковые существуют);
- vl — левый ребенок v , vr — правый ребенок v ;
- pl — левый ребенок p , pr — правый ребенок p .

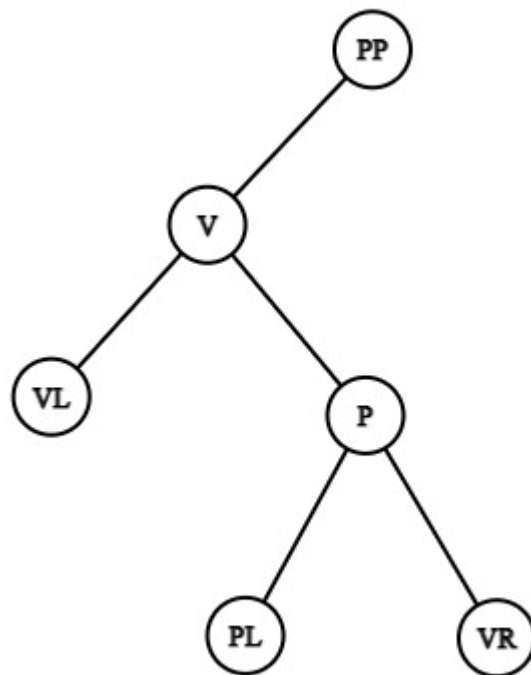
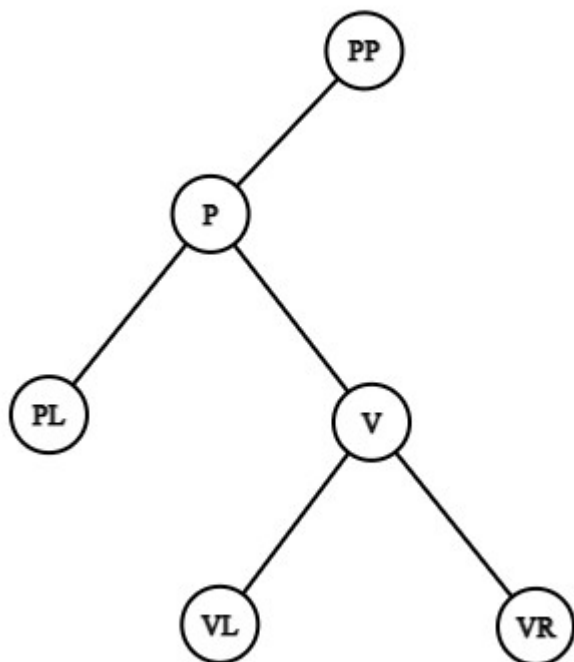
В таком случае обмен задаётся следующими условиями:

- **Изменение предка:** если p — левый ребенок вершины pp , то v становится левым ребенком pp , иначе — правым.
- если v — **левый ребенок** вершины p , то:
 1. p становится левым ребенком v ;
 2. vr остаётся правым ребенком v ;
 3. vl становится левым ребенком p ;
 4. pr остаётся правым ребенком p .



● аналогично, если v — **правый ребенок** вершины p , то:

1. p становится правым ребенком v ;
2. vl остаётся левым ребенком v ;
3. vr становится правым ребенком p ;
4. pl остаётся левым ребенком p .



Формат ввода

Первая строка содержит два целых числа N и Q ($1 \leq N \leq 750$; $1 \leq Q \leq 106$) — количество вершин в дереве и количество изменений, произошедших с деревом.

В следующей строке дано Q целых чисел v_1, v_2, \dots, v_q ($1 \leq v_i \leq N$), где v_i — номер вершины, обменявшейся местами со своим предком в i -ю ночь.

Формат вывода

В единственной строке через пробел требуется вывести номера вершин дерева после всех изменений в формате LVR, начиная с корня дерева.

Формат LVR(v) определяется рекурсивно для вершины v .

- если у вершины v есть левый ребенок lv , то сначала выводится всё поддерево lv в формате LVR(lv);
- выводится номер вершины v ;
- если у вершины v есть правый ребенок rv , то выводится всё поддерево rv в формате LVR(rv);

Пример

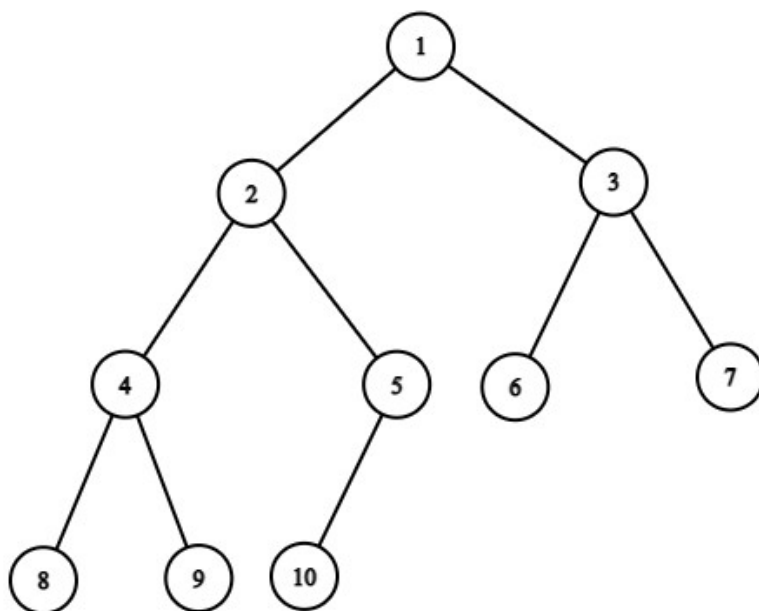
Ввод	Вывод
10 6	
5 7 4 7 8 7	7 10 5 2 8 4 9 1 6 3

Примечания

Объяснение примера строится из двух частей:

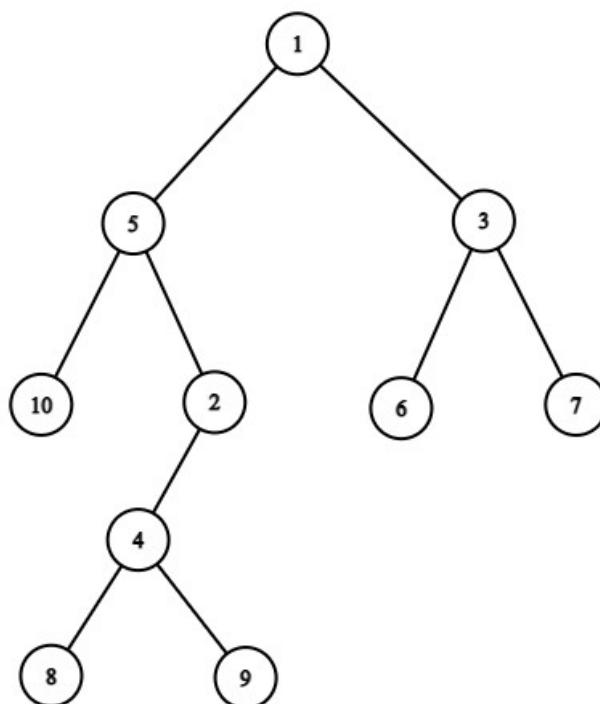
- Рисунки, показывающие структуру дерева в каждом сне;
- Текстовое пояснение к выводу на тест;

В тестовом примере дано дерево из 10 вершин:

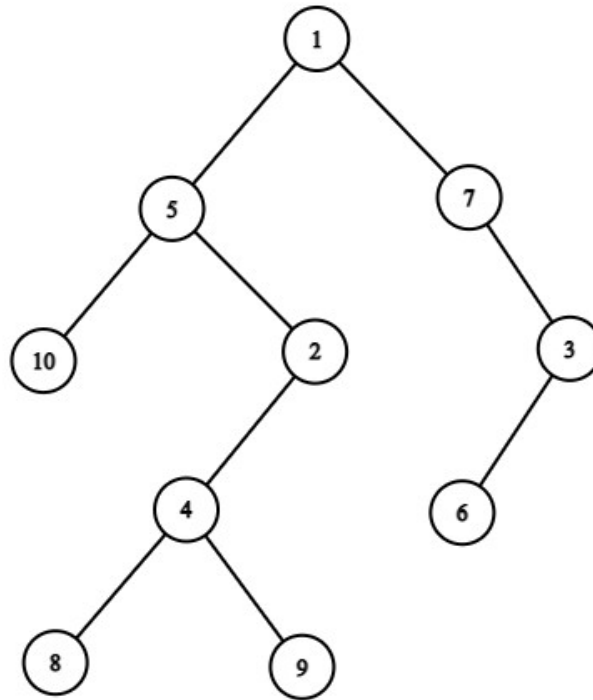


Происходит 6 изменений с деревом:

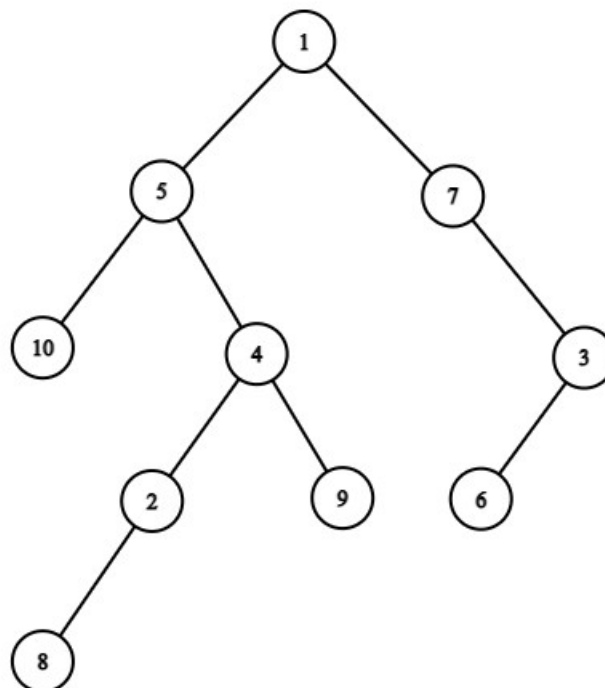
1. вершина 5 меняется с вершиной 2:



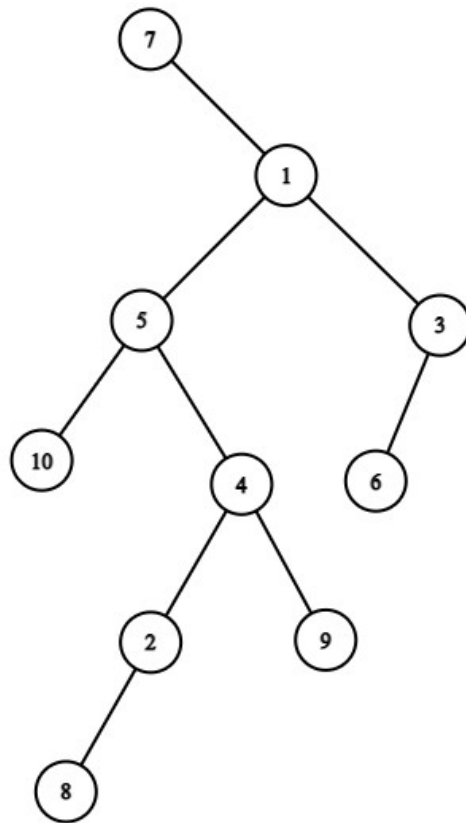
2. вершина 7 меняется с вершиной 3:



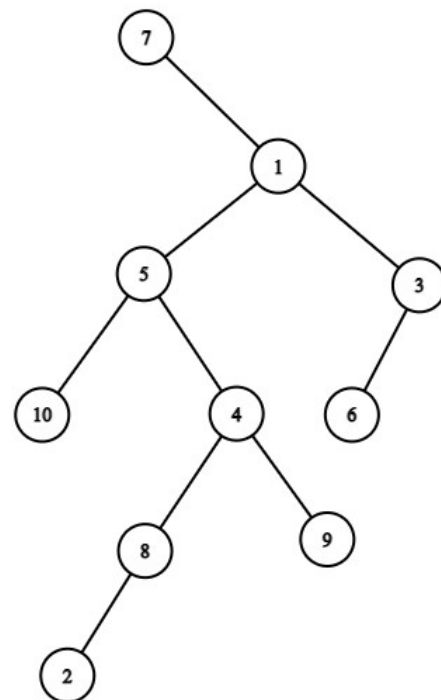
3. вершина 4 меняется с вершиной 2:



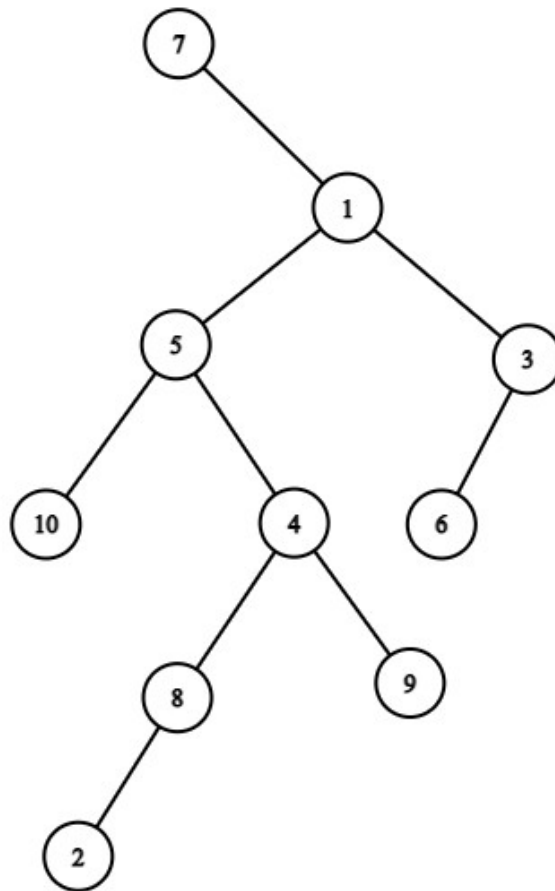
4. вершина 7 меняется с вершиной 1:



5. вершина 8 меняется с вершиной 2:



6. вершина 7 ни с кем не меняется, так как она уже корень:



После всех изменений выводим получившееся дерево в формате LVR:

- Корень дерева — вершина 7;
- **Выводим вершину 7;**
- Выводим поддереву вершины 1 — правого ребенка вершины 7;
- Выводим поддереву вершины 5 — левого ребенка вершины 1;
- Выводим поддереву вершины 10 — левого ребенка вершины 5;
- **Выводим вершину 10;**
- **Выводим вершину 5;**
- Выводим поддереву вершины 4 — правого ребенка вершины 5;
- Выводим поддереву вершины 8 — левого ребенка вершины 4;
- Выводим поддереву вершины 2 — левого ребенка вершины 8;
- **Выводим вершину 2;**
- **Выводим вершину 8;**
- **Выводим вершину 4;**
- Выводим поддереву вершины 9 — правого ребенка вершины 4;
- **Выводим вершину 9;**
- **Выводим вершину 1;**
- Выводим поддереву вершины 3 — правого ребенка вершины 1;
- Выводим поддереву вершины 6 — левого ребенка вершины 3;
- **Выводим вершину 6;**
- **Выводим вершину 3;**

D. Лей, лей, не жалеЙ

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	3 секунды	512Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Python 3.7.3	10 секунд	512Mb		
Python 3.7 (PyPy 7.3.3)	10 секунд	512Mb		
Scala 2.13.4	6 секунд	512Mb		
OpenJDK Java 15	6 секунд	512Mb		
PHP 7.3.5	6 секунд	512Mb		
Kotlin 1.5.32 (JRE 11)	6 секунд	512Mb		

В известной компании Тындекс уже несколько лет работает очень популярный сервис Тындекс.Вода, занимающийся поливом людей, растений, зданий и всего остального, что можно полить без ущерба окружающим.

Пользователь указывает, что ему необходимо полить и сколько литров воды он готов на это потратить, после чего на место выезжает специальная бригада. В итоге для компании каждый заказ можно представить тройкой чисел:

- время `Start`, когда бригада приняла заказ и выехала;
- время `End`, когда бригада осуществила заказ и освободилась;
- итоговая стоимость заказа `Cost`.

Для простоты обработки и хранения время задается одним целым числом, равным количеству минут, прошедших с начала запуска сервиса до искомого момента.

Продолжительность заказа считается равной величине `End–Start`.

Начальнику сервиса необходимо отчитаться перед вышестоящим начальством, поэтому он поручил вам несложную задачу — найти ответы на несколько запросов одного из двух типов:

1. Найти суммарную стоимость заказов, которые начались в заданный промежуток времени;
2. Найти суммарную продолжительность заказов, которые завершились в заданный промежуток времени;

В обеих статистиках промежутки считаются **отрезками**: в промежуток от `Start` до `End` входят все величины `Start`, `Start+1`, ..., `End–1`, `End`.

Формат ввода

В первой строке расположено одно целое число N ($1 \leq N \leq 200000$) — количество заказов, осуществленных сервисом.

Каждая из следующих N строк содержит информацию об одном заказе в формате `StartEndCost` ($1 \leq \text{Start} < \text{End} \leq 10^9$; $1 \leq \text{Cost} \leq 10^9$) — время начала и конца заказа и стоимость заказа соответственно.

В следующей строке расположено одно целое число Q ($1 \leq Q \leq 200000$) — количество запросов.

Каждая из следующих Q строк содержит информацию об одном запросе в формате `StartEndType` ($1 \leq \text{Start} \leq \text{End} \leq 10^9$; $1 \leq \text{Type} \leq 2$) — время начала и конца промежутка и тип запроса соответственно.

Соответствие типов запроса следующее:

1. Найти суммарную стоимость заказов, которые начались в заданный промежуток времени;
2. Найти суммарную продолжительность заказов, которые завершились в заданный промежуток времени;

Формат вывода

В единственной строке через пробел выведите Q целых чисел — ответы на запросы в порядке их ввода.

Пример 1

Ввод	Вывод
1	
10 100 1000	
6	
1 10 1	
1 10 2	1000 0 1000 90 0 90
10 100 1	
10 100 2	
100 1000 1	
100 1000 2	

Пример 2

Ввод	Вывод
5	
5 20 5	
6 21 4	
6 22 3	
7 23 2	
10 24 1	10 12 3
3	
6 11 1	
4 6 1	
7 11 1	

Пример 3

Ввод	Вывод
7	5 14 14 28 13 22
3 6 1	
4 6 2	
3 4 3	
4 10 100500	
4 11 777	
3 8 365	
4 8 31	
6	
6 6 2	
6 8 2	
5 9 2	
3 12 2	
9 12 2	
8 12 2	

Примечания

Первый тестовый пример.

Есть данные про 1 заказ:

1. с 10-й по 100-ю минуту стоимостью 1000;

Необходимо ответить на следующие 6 запросов:

1. суммарная стоимость заказов, начавшихся в промежутке между 1-й и 10-й минутами;
2. суммарная продолжительность заказов, закончившихся в промежутке между 1-й и 10-й минутами;
3. суммарная стоимость заказов, начавшихся в промежутке между 10-й и 100-й минутами;
4. суммарная продолжительность заказов, закончившихся в промежутке между 10-й и 100-й минутами;
5. суммарная стоимость заказов, начавшихся в промежутке между 100-й и 1000-й минутами;
6. суммарная продолжительность заказов, закончившихся в промежутке между 100-й и 1000-й минутами.

Единственный в тесте заказ подходит под:

- первый запрос, так как начало запроса 10 удовлетворяет условию $1 \leq 10 \leq 10$;
- третий запрос, так как начало запроса 10 удовлетворяет условию $10 \leq 10 \leq 100$;
- четвертый запрос, так как конец запроса 100 удовлетворяет условию $10 \leq 100 \leq 100$;
- шестой запрос, так как конец запроса 100 удовлетворяет условию $100 \leq 100 \leq 1000$;

Второй тестовый пример.

Есть данные про 5 заказов:

1. с 5-й по 20-й минуту стоимостью 5;
2. с 6-й по 21-ю минуту стоимостью 4;
3. с 6-й по 22-ю минуту стоимостью 3;
4. с 7-й по 23-ю минуту стоимостью 2;
5. с 10-й по 24-ю минуту стоимостью 1.

Необходимо ответить на следующие 3 запроса про суммарную стоимость заказов, начавшихся в заданном промежутке:

1. между 6-й и 11-й минутами;
2. между 4-й и 6-й минутами;
3. между 7-й и 11-й минутами;

Под первый запрос подходят заказы 2, 3, 4, 5, поэтому ответом на запрос будет их суммарная стоимость $4 + 3 + 2 + 1 = 10$.

Под второй запрос подходят заказы 1, 2 и 3 — поэтому ответом будет $5 + 4 + 3 = 12$.

Третьему запросу удовлетворяют лишь заказы 4, 5, поэтому ответом на запрос будет $2 + 1 = 3$.

Третий тестовый пример.

Есть данные про 7 заказов:

1. с 3-й по 6-ю минуту стоимостью 1;
2. с 4-й по 6-ю минуту стоимостью 2;
3. с 3-й по 4-ю минуту стоимостью 3;
4. с 4-й по 10-ю минуту стоимостью 100500;
5. с 4-й по 11-ю минуту стоимостью 777;
6. с 3-й по 8-ю минуту стоимостью 365;
7. с 4-й по 8-ю минуту стоимостью 31.

Необходимо ответить на следующие 6 запросов про суммарную продолжительность заказов, закончившихся в заданном промежутке:

1. между 6-й и 6-й минутами;
2. между 6-й и 8-й минутами;
3. между 5-й и 9-й минутами;
4. между 3-й и 12-й минутами;
5. между 9-й и 12-й минутами;
6. между 8-й и 12-й минутами;

Под первый запрос подходят заказы 1 и 2 — их суммарная продолжительность $(6-3)+(6-4)=3+2=5$.

Под второй запрос подходят заказы 1, 2, 6 и 7 — их суммарная продолжительность равна $(6-3)+(6-4)+(8-3)+(8-4)=3+2+5+4=14$.

Третьему запросу удовлетворяют те же самые заказы, что и под второй — поэтому ответ также равен **14**.

Четвертый запрос включает в себя вообще все заказы, поэтому ответ на данный запрос равен $(6-3)+(6-4)+(4-3)+(10-4)+(11-4)+(8-3)+(8-4)=3+2+1+6+7+5+4=28$.

В пятом запросе рассматриваются заказы 4 и 5 — ответом будет $(10-4)+(11-4)=6+7=13$.

Последний, шестой запрос затрагивает заказы 4, 5, 6 и 7 — их суммарная продолжительность равна $(10-4)+(11-4)+(8-3)+(8-4)=6+7+5+4=22$.

Е. Занимательная алхимия

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	2 секунды	512Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Python 3.7.3	5 секунд	512Mb		
Python 3.7 (PyPy 7.3.3)	5 секунд	512Mb		
Scala 2.13.4	5 секунд	512Mb		
PHP 7.3.5	5 секунд	512Mb		
OpenJDK Java 15	5 секунд	512Mb		
Kotlin 1.5.32 (JRE 11)	5 секунд	512Mb		

На рынке зелий произошёл бум и практически в каждом доме появилась своя алхимическая лаборатория. Ресурсы подобных индивидуальных предпринимателей невелики, и их инструментарий сильно ограничен. В результате долгих экспериментов были открыты два универсальных ингредиента для зелий (назовем их А и В), которые оказались доступны любому начинающему алхимику.

Отныне рецепт любого зелья можно свести к этим двум компонентам - зелье состоит из чистых ингредиентов А и/или В, из других зелий или из смеси чистых ингредиентов и зелий.

Одно зелье может требовать несколько экземпляров одного и того же ингредиента / зелья.

Ваш близкий друг Тирания Вампадур только начинает свой путь в алхимии и каждый день задаёт вам Q вопросов вида: если в её подвале осталось X единиц ингредиента А и Y единиц ингредиента В, может ли она изготовить один пузырёк зелья номер S?

Для облегчения своей жизни, вы решаете написать программу, отвечающую на подобные вопросы и подарить её Тирании.

Важно отметить, что некоторые рецепты были записаны со слов почётных алхимиков города, чья память знавала и лучшие времена. Поэтому рецепты для некоторых зелий могут быть записаны неправильно и содержать циклы — такие зелья изготовить никак нельзя.

Формат ввода

В первой строке дано число N ($3 \leq N \leq 200000$) — общее количество ингредиентов и рецептов производных зелий.

Ингредиент А имеет номер 1, В — номер 2, все производные зелья пронумерованы от 3 до N.

Следующие N-2 строк содержат информацию о создании зелий: в i-й строке содержится список составных частей для зелья i+2.

Первое число в строке K_i — количество составных частей. Далее через пробел следуют K_i чисел P_{ij} ($1 \leq P_{ij} \leq N$, $1 \leq j \leq K_i$) — номера составных частей для изготовления зелья i+2.

Составные части в строке могут повторяться — каждая часть учитывается столько раз, сколько указана.

Гарантируется, что $\sum K_i \leq 1000000$.

В следующей строке задано одно целое число Q ($1 \leq Q \leq 200000$) — количество вопросов Тирании.

Каждый вопрос задаётся в отдельной строке в формате XYS ($0 \leq X, Y \leq 10^9$, $3 \leq S \leq N$) — количество ингредиентов А и В в подвале соответственно, а также номер запрашиваемого для изготовления зелья.

Формат вывода

В единственной строке выведите строку из Q символов:

- 1, если можно изготовить зелье из имеющегося количества ингредиентов;
- 0 в любом ином случае.

Пример

Ввод	Вывод
------	-------

7	
3 1 1 2	
2 1 3	
3 4 3 4	
1 7	100
1 6	
3	
8 4 5	
9 2 5	
10 10 6	

Примечания

В приведенном тестовом примере есть 5 зелий и 2 чистых ингредиента:

- Зелье 3 требует для создания два ингредиента А и один ингредиент В;
- Зелье 4 требует один ингредиент А и одно зелье 3 — значит суммарно три ингредиента А и один ингредиент В;
- Зелье 5 требует два зелья 4 и одно зелье 3 — суммарно восемь ингредиентов А и три ингредиента В;
- Зелья 6 и 7 ссылаются друг на друга в рецептах — судя по всему, что-то перепуталось при записи и в данный момент изготовить их никак нельзя.

В первом вопросе Тиранию интересует, можно ли изготовить зелье 5, имея ровно 8 ингредиентов А и 4 ингредиента В. Как мы выяснили выше, для изготовления зелья 5 требуется как раз 8 ингредиентов А и 3 ингредиента В — значит зелье изготовить можно.

Второй вопрос Тирания задаёт также про зелье 5, но теперь ингредиентов А — целых 9, в то время как ингредиентов В всего 2. Хотя ингредиентов А хватает с избытком, но ингредиентов В не хватает (нужно 3), поэтому изготовить зелье нельзя.

В третий раз Тирания спрашивает вас о зелье 6 — вы уже знаете, что независимо от количества ингредиентов это зелье изготовить не получится никак из-за циклической зависимости в рецептуре.