

Тренировочный контест — бэкенд

C. Приснится же такое...

✓ Полное решение

	Все языки	C++20 (GCC 14.1)	C++20 (Clang 18.1.6)
Ограничение времени	3 секунды	2 секунды	2 секунды
Ограничение памяти	512Mb	512Mb	512Mb
Ввод	стандартный ввод или input.txt		
Вывод	стандартный вывод или output.txt		

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 двойичное 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 ночных снегов 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.

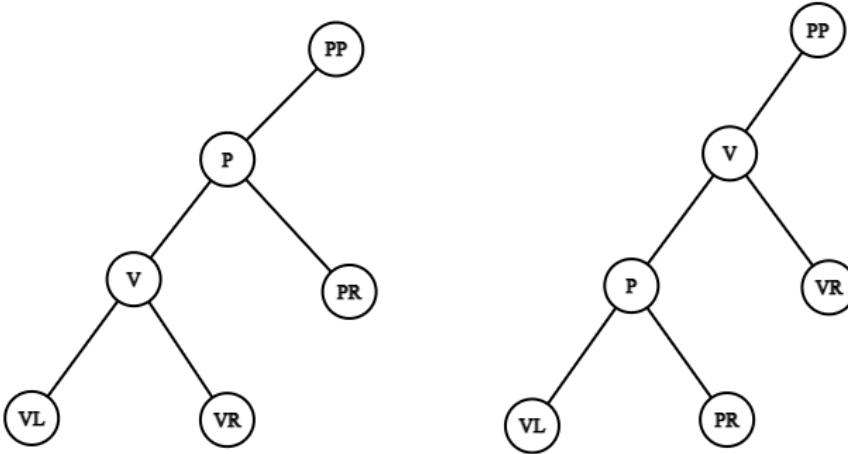
Понимая, что в этом деле важна точность, вы расспросили Тиранию насчет процесса обмена местами вершины v с её предком.

Введем обозначения:

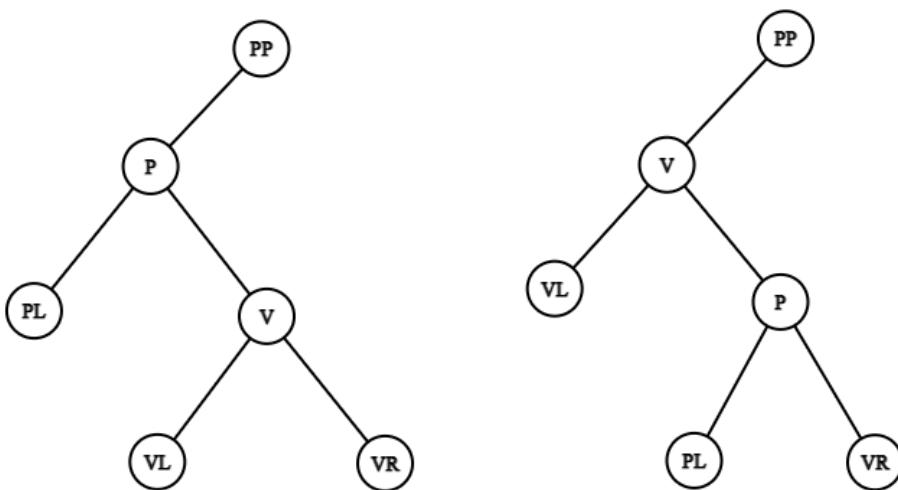
- p — предок вершины v , pp — предок вершины p (если такие существуют);
- vl — левый ребенок v , vr — правый ребенок v ;
- pl — левый ребенок p , pr — правый ребенок p .

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:
 - p становится правым ребенком v ;
 - vl остается левым ребенком v ;
 - vr становится правым ребенком p ;
 - pl остается левым ребенком p .



Формат ввода

Первая строка содержит два целых числа N и Q ($1 \leq N \leq 750$; $1 \leq Q \leq 10^6$) — 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

10 6
5 7 4 7 8 7

Выход

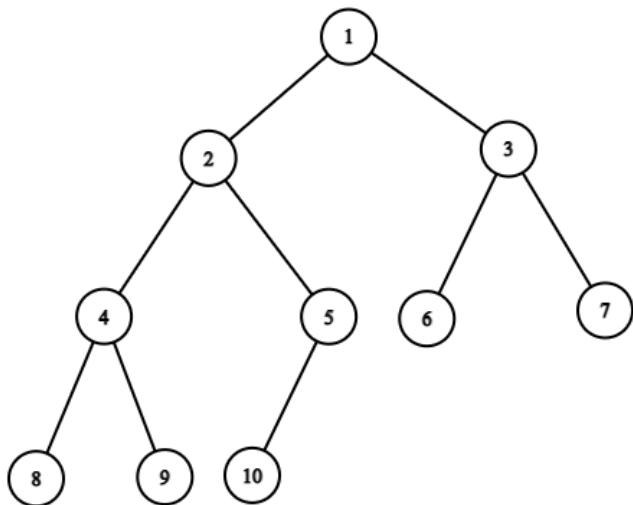
7 10 5 2 8 4 9 1 6 3

Примечания

Объяснение примера строится из двух частей:

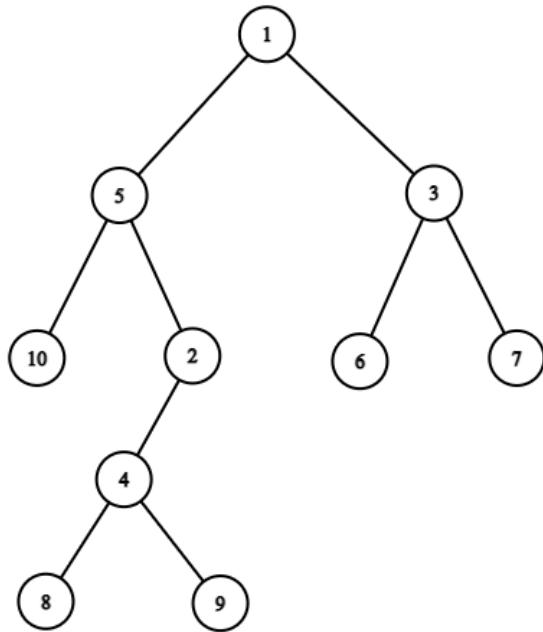
- Рисунки, показывающие структуру дерева в каждом сне;
- Текстовое пояснение к выводу на тест;

В тестовом примере дано дерево из 10 вершин:

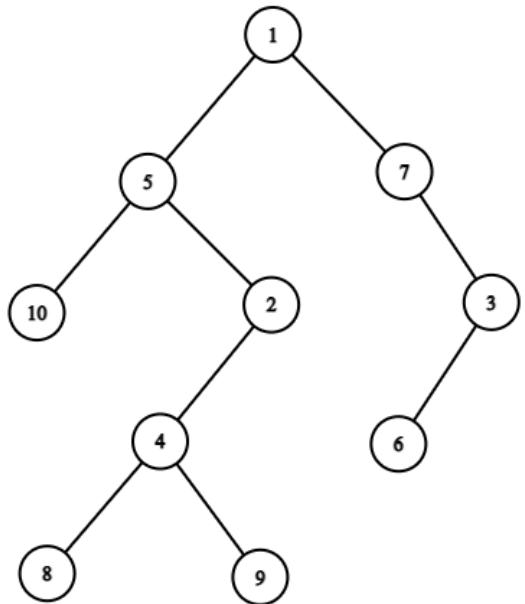


Происходит 6 изменений с деревом:

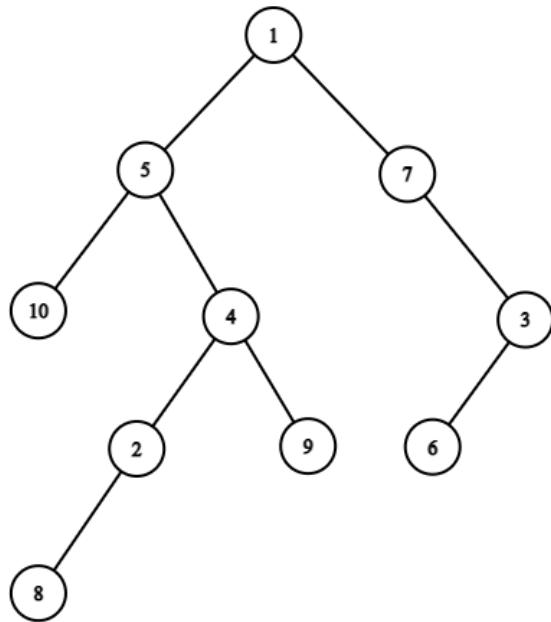
1. вершина 5 меняется с вершиной 2:



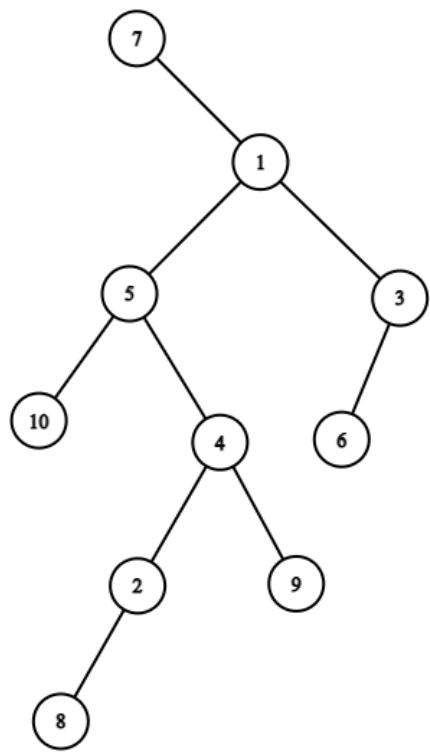
2. вершина 7 меняется с вершиной 3:



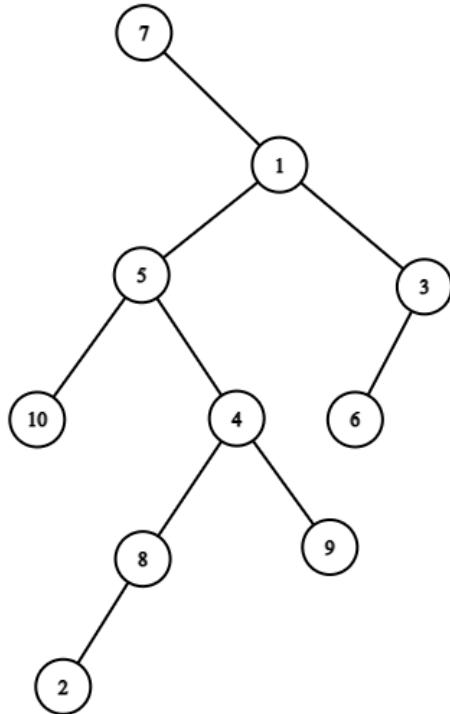
3. вершина 4 меняется с вершиной 2:



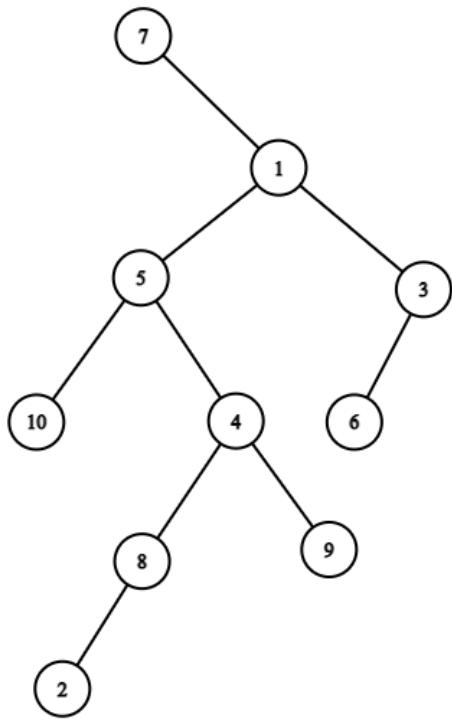
4. вершина 7 меняется с вершиной 1:



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