Агент 007700 разполага с дърво с $n$ върха и корен връх с номер $1$. Във всеки връх има стойност - $x\_{1}, x\_{2}, … , x\_{n}$. За негово съжаление, агентът на ДпС (Департамент по Сигурността) Божил се е добрал до него с цел да саботира. Случват се събития от 2 типа:

1. $Update$ с параметри $node и value$. Тогава агент Божил е променил $x\_{node}$ и то е станало равно на $value$
2. $Query$ с параметър $node$. Тогава агент 007700 иска да намери побитово “изключващо или“ (xor) на стойностите от $node$ и всички негови преки и непреки предшественици

**Вход**

От първия ред на файла **xor.in** се въвеждат числата $n$ и $q$. На следващия ред са дадени $n$ числа - $x\_{1}, x\_{2}, … , x\_{n}$ – стойностите по върховете. Следват $n-1$ реда по 2 естествени числа - $u, v$ – ребрата в дървото. Следват $q$ на брой реда, всеки започва с 2 числа - $type и node$.

Ако $type=1$, това е операция $Update$ и се въвежда още 1 число - $value$

Ако $type=2$, това е операция $Query$

**Изход**

За всяка заявка от тип 2 отпечатайте на нов ред във файла **xor.out** търсената стойност за съответната заявка.

**Ограничения**

$$1\leq n\leq 20 000$$

$$1\leq q\leq 50 000$$

$$1\leq x\_{i}, value\leq 10^{9}$$

$$1\leq type\leq 2$$

$$1\leq node\leq n$$

 **Ограничение по време: 0.7 sec.**

 **Ограничение по памет: 256 MB.**

**Примерен тест**

|  |  |
| --- | --- |
| **Вход (xor.in)** | **Изход (xor.out)** |
| 4 31 2 4 81 22 33 42 41 2 52 3 | 150 |