Graft and Transplant

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Alice and Bob have a tree with n vertices numbered from 1 to n, and they want to play a game on this tree to determine the only winner. They decide to perform the following operation alternately, with Alice going first. In each operation, they can choose two adjacent vertices u and v, and make all the undirected edges connecting u, except the one connecting u and v, reconnect to v instead of u. In short, each edge (u, w) where $w \neq v$ changes to (v, w) after the operation.

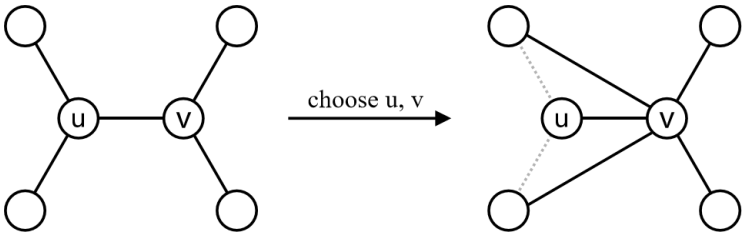

Figure: The "Graft and Transplant" Operation

However, they soon realize that such an operation will probably result in a game that never ends, so they add an extra rule that the two trees before and after an operation must not be isomorphic. More formally, let $V(S)$ be the set of vertices in a tree S and $V(T)$ be the set of vertices in a tree T. Tree S and tree T are *isomorphic* if there exists a bijection $f : V(S) \to V(T)$ such that for all pairs of vertices (u, v) in $V(S)$, u and v are connected by an edge in S if and only if vertices $f(u)$ and $f(v)$ are connected by an edge in T. Namely, $f(s) = t$ implies vertex s in S corresponds to vertex t in T.

In this scenario, when one player cannot perform any valid operations, the other player will win the game. Assuming both Alice and Bob are clever enough and will adopt the best strategy to win or prevent the opponent from winning if one is unable to win, you need to predict the winner.

Input

The first line contains an integer n ($2 \leq n \leq 50$), denoting the number of vertices in the tree.

Then $n-1$ lines follow, each of which contains two integers u and v ($1 \leq u, v \leq n$), denoting an undirected edge connecting vertices u and v. It is guaranteed that the given edges form a tree.

Output

Output "`Alice`" if Alice will win the game, "`Bob`" if Bob will win the game, or "`Draw`" if the game under the extra rule will still never end if both of them perform their operations optimally.

Examples

standard input	standard output
4 1 2 2 3 3 4	Alice
4 1 2 1 3 1 4	Bob