

Submit: xmas.c / xmas.cpp / xmas.pas  
 Vstup: knihovna  
 Výstup: knihovna

Časový limit: sekundy  
 Paměťový limit: 64 MB  
 Bodů: 100

Koupil jsi krásný vánoční stromek (pravda, nejsou zrovna Vánoce, ale byl ve slevě) a krabici vánočních ozdob čtyř druhů: Baňky, Duté figurky, Sněhuláky a Andělíčky.

Stromeček je tvořen mnoha místy, kam se dá ozdoba umístit. Některá místa jsou spojena větví. Celý strom je z jedné části (všechna místa jsou vzájemně propojena) a nejsou v něm žádné kružnice.

Vsadiš se se svým přítelem. Tvrdil jsi, že ti nedokáže zabránit v ozdobení stromečku legálním způsobem. (Neříkej, že jsi nevěděl, že máme zákon upravující zdobení stromečků) Strom je ozdoben legálně, pokud žádná dvě místa přímo spojená větví nemají ozdobu stejného typu.

Budete se střídát v tazích (začínáš), každý vždy ozdobí jedno místo. Nikdo z vás nemůže umístit ozdobu se stejným typem, jako má některý ze sousedů místa, kam je ozdoba umísťována (bylo by to protizákonné).

Pokud se ti povede dosáhnout stavu, kdy jsou všechna místa ozdobena, vyhráváš. Pokud ne, pak prohráváš.

### Herní knihovna:

Tato úloha nemá žádné vstupní ani výstupní soubory. K dispozici máš knihovnu, která ti podá informaci o struktuře stromu a bude simulovat nepřítele.

Knihovna se nazývá `tree_lib` a do programu ji přidáš buď pomocí `#include "tree_lib.h"` nebo `use tree_lib;` (záleží na použitém jazyku). Knihovna poskytuje tyto funkce:

- `int init( void );`  
`function init: longint;`  
 Tuto funkci musíš zavolat právě jednou, před tím než zavoláš cokoli jiného z knihovny. Vrátí  $N$ —počet ozdobitelných míst na stromu. Od teď bude mít každé takové místo číslo z intervalu  $1, \dots, N$ .
- `int neighbor_count( int point );`  
`function neighbor_count( point: longint ): longint;`  
 Tato funkce ti řekne, kolik má místo sousedů. Pokud je parametr *point* mimo rozsah, vrátí 0.
- `int neighbor( int point, int index );`  
`function neighbor( point, index: longint ): longint;`  
 Zavoláním této funkce se dozvíš číslo *index*-tého souseda místa *point*. Pokud je některý z parametrů mimo rozsah, vrátí 0. (První soused má index 1).
- `char decoration( int point );`  
`function decoration( point: longint ): char;`  
 Tato funkce vrátí ozdobu místa *point*, což je jedna z hodnot 'A', 'B', 'D' nebo 'S' pro andělíčka, baňku, dutou figurku nebo sněhuláka, nebo ' ' pro neozdobené místo nebo neexistující místo.
- `int enemy( void );`  
`function enemy: longint;`  
 Tato funkce může být zavolána, jen když je nepřítel na tahu. Vrátí číslo místa, které nepřítel ozdobil. Zjištění typu ozdoby je již na tobě.
- `void decorate( int point, char decoration );`  
`procedure decorate( point: longint; decoration: char );`  
 Použij toto k ozdobení místa *point* typem *decoration* (jedna z 'A', 'B', 'D' nebo 'S'). Toto můžeš zavolat jen, když jsi na tahu.  
 Pokud se pokusíš ozdobit neexistující nebo již ozdobené místo, umístit ozdobu protizákonně, předáš nepříteli strom, kde jsou ještě neozdobená místa, ale žádné již nesmí být legálně ozdobeno, nebo nedodržíš pořadí hráčů, program bude ukončen a prohraješ. Když je ozdobený celý strom, program je taktéž ukončen (ve funkci, která toto způsobila), tentokrát však vyhráváš.