

## Csillag molekula központja

Kutatók egy speciális molekulát vizsgálnak. Tudják, hogy a molekula  $N$  különböző atomot tartalmaz, amelyek csillag szerkezetet alkotnak. Ez azt jelenti, hogy pontosan egy olyan atom van a molekulában, amelyhez legalább három atom kapcsolódik közvetlenül, és a többi atomhoz legfeljebb kettő. Természetesen a molekula összefüggő, azaz bármely két atomja között van közvetlen szomszédokon keresztül vezető útvonal. A kutatók a molekulát egy speciális mérőműszerrel vizsgálják. A műszer bármely két adott atomjára megad egy közbülső atomot.

Írj programot, amely meghatározza a molekula központi atomját, tehát azt, amelyhez legalább három atom kapcsolódik közvetlenül!

A programod nem írhat, és nem olvashat semmilyen fájlt, beleértve a standard inputot és outputot is!

### Könyvtári műveletek

A mérőműszer használatát a `muszer` könyvtár három művelete biztosítja:

`Atomszam`, egyszer kell hívni a program elején, az atomok ( $7 \leq N \leq 1000$ ) számát adja. Az atomokat az  $1, \dots, N$  számokkal azonosítjuk.

`Kozte`, két atom azonosítóját kell argumentumként megadni `Kozte(x, y)`; a visszaadott érték egy olyan atom azonosítója, amely a molekulában az  $x$  és  $y$  atom között van. Ha  $x$  és  $y$  nem közvetlen szomszédok a molekulában, akkor a visszaadott érték biztosan nem  $x$  és nem  $y$ . Ha  $x$  és  $y$  közvetlen szomszédok, akkor a visszaadott érték vagy  $x$ , vagy  $y$ .

`Kozpont`, a program végén kell hívni, a kiszámított eredmény közléséhez! Az argumentum a központi atom sorszáma legyen!

### Gyakorlás

Gyakorlás. Letölthető egy minta `muszer` modul C++ és Pascal forrás programja. A `muszer` a standard bemenetről egy egész számot olvas be, az atomok  $N$  számát. A válaszokat egy véletlenszerű elrendezés alapján adja.

Pascal program esetén:

```
uses muszer;
```

A műveletek Pascal deklarációja

```
function Atomszam:integer;  
function Kozte(x, y : integer) : integer;  
procedure Kozpont(x:integer);
```

A műveletek C/C++ deklarációja

```
#include "muszer.h"  
int Atomszam(void);  
int Kozte(int x, int y);  
void Kozpont(int x);
```

### Korlátok

Időlimit: 1.0 mp.

Memórialimit: 32MB

Pontozás: Ha a Központ eljárással közölt atom helyes és legfeljebb  $2*N+10$  Kozte műveletet hajtott végre a program, akkor teljes jár. Ha legfeljebb  $3*N$  Kozte műveletet hajtott végre a program, akkor 50% pont jár. A tesztek 30%-ában  $N \leq 100$ .

