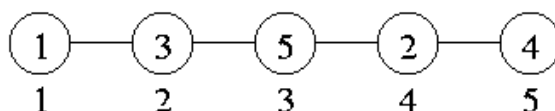


Molekula

Kutatók egy speciális molekulát vizsgálnak. Tudják, hogy a molekula N különböző atomot tartalmaz, amelyek egy lineáris láncot alkotnak. A kutatók a molekulát egy speciális mérőműszerrel vizsgálják. A műszer meg tudja adni a molekula két adott atomja közötti egyik olyan atomot, amely a két atom valamelyikének közvetlen szomszédja. Például az ábrán szereplő molekula esetén ha az 1-es és 2-es atomra kérdeznek, akkor a válasz 3, vagy 5 lesz.

Írj programot, amely meghatározza a molekula szerkezetét, azaz minden atom pozícióját a molekulában! A megoldás tükörkép erejéig egyértelmű.

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 számát ($2 \leq N \leq 10\,000$) adja.

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

`Sorrend(i, x)`, a program végén kell hívni, annak közléséhez, hogy a molekulában az i -edik pozíción a x azonosítójú atom van. Minden i -re ($1 \leq i \leq N$) pontosan egyszer kell hívni, az utolsó hívás esetén az első paraméter értéke N legyen, ekkor ugyanis a program végrehajtása befejeződik! A megoldás tükörkép erejéig egyértelmű, a két megoldás közül bármelyiket meg lehet adni.

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;

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

A műveletek C/C++ deklarációja

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

Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32MB

Pontozás: Ha a Sorrend eljárással közölt sorrend helyes és legfeljebb $3 \cdot N$ Kozte műveletet hajtott végre a program, akkor 1 pont/teszteset jár, ha legfeljebb $2 \cdot N$ Kozte műveletet hajtott végre a program, akkor teljes pontszám jár. A tesztek 40%-ában $N \leq 400$.