

Vám

Egy megye településeit úthálózat köti össze. Az egyes települések a bevételeik növelése érdekében minden, rajtuk áthaladó autótól vámot szednek. Időnként a megyében új utak is épülnek.

Készíts programot, amely megadja egyes városok közötti fizetendő vám összegét, illetve adminisztrálja az új utak létrehozását!

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

Könyvtári műveletek

A vámok kezelését a vamok könyvtár hat művelete biztosítja:

Telepulesszam, egyszer kell hívni a program elején, a települések számát ($2 \leq N \leq 250$) adja.

Utszam, egyszer kell hívni az előző függvény után, a települések közötti közvetlen utak számát ($0 \leq U \leq N * N$) adja.

Vam(i), minden i -re meg kell hívni a program elején, az i . településen áthaladó autók által fizetendő vámot adja ($1 \leq \text{Vam}(i) \leq 100$).

VanUt(i, j), U -szor kell hívni a program elején, i és j település sorszámot ad, amelyek között van közvetlen, kétirányú útszakasz ($1 \leq i, j \leq N$).

Igeny, egy kérdést tartalmaz Igeny(t, x, y); ha t értéke 1, akkor az x és y város közötti úton fizetendő s vámot kell megadni az ezt követő Utvam(s) eljárással; ha t értéke 2, akkor az azt jelenti, hogy kiépült egy út az x és y város között ($1 \leq x, y \leq N$). Ha nem lehet eljutni x -ből y -ba, akkor az Utvam eljárást -1 paraméterrel kell meghívni! A $t=2$ esetén elképzelhető, hogy x és y között volt már másik út. Az Igeny eljárást legfeljebb 100 000-szer hívhatod meg, $t=2$ értéket legfeljebb 1000-szer kapsz. Az Igeny eljárás befejezi a programod futását, ha nincs több igény.

Gyakorlás

Gyakorlás. Letölthető egy minta vamok modul C++ és Pascal forrás programja. A vamok a standard bemenet első sorából olvassa be a települések és a közöttük levő útszakaszok számát, a második sorban pedig az egyes településeken fizetendő vámokat. A következő sorokban két-két település sorszám van, amelyek között van közvetlen útszakasz. A következő sorokban található t, x, y értéke. Az utolsó sorban három darab 0 jelzi az igények végét.

Pascal program esetén:

```
uses vamok;
```

A műveletek Pascal deklarációja

```
function Telepulesszam: integer;  
function Utszam: integer;  
function Vam(i: integer): integer;  
procedure VanUt(var x, y: integer);  
procedure Igeny(var t, x, y: integer);  
procedure Utvam(s: integer);
```

A műveletek C/C++ deklarációja

```
#include "vamok.h"  
int Telepulesszam(void);  
int Utszam(void);  
int Vam(int i);
```

```
void VanUt(int &x, int &y);  
void Igeny(int &t,int &x, int &y);  
void Utvam(int s);
```

Korlátok

Időlimit: 1.0 mp.

Memórialimit: 32MB

Pontozás: A tesztek 30%-ában $N \leq 10$.