

## Kavicsos játék

Népszerű az a kétszemélyes játék, amelyet kavicsokkal játszanak. A játék kezdetén letesznek két kupacba kavicsokat. A játékosok felváltva lépnek. Egy lépésben el lehet venni vagy egy kupacból akárhány, vagy mindkettőből ugyanannyi kavicsot. Az nyer, aki az utolsó kavicsot elveszi. A bemenetek mindegyike esetén a kezdő játékosnak van nyerő stratégiája.

Írj programot, amely a kezdő játékos nyerő játékát valósítja meg!

### Könyvtár

Az alábbi műveleteket lehet használni, amelyeket az `Ellenfel` modul valósít meg.

- `KavicsokSzama(m, n)`: Pontosan egyszer kell hívni a program elején. Az  $m$  változóban az első, az  $n$  változóban pedig a második kupacban lévő kavicsok számát adja ( $3 \leq m, n \leq 10000$ ).
- `EnLepesem(k1, k2)`: Az első játékos lépését valósítja meg; ami azt jelenti, hogy az első kupacból  $k1$ , a másodikból  $k2$  kavicsot vesz el.
- `TeLepesed(k1, k2)`: A második játékos (az ellenfél) lépését adja meg; ami azt jelenti, hogy az első kupacból  $k1$ , a másodikból  $k2$  kavicsot vett el.
- Az utolsó lépés végrehajtása a program befejeződését eredményezi.

### Az `Ellenfel` modul használata.

Pascal program esetén

```
uses Ellenfel;
```

Műveletek Pascal deklarációja

```
procedure KavicsokSzama(var m, n:integer);  
procedure EnLepesem(k1, k2 : integer);  
procedure TeLepesed(var k1, var k2 : integer);
```

A műveletek C/C++ deklarációja

```
#include "Ellenfel.h"  
void KavicsokSzama(int &m, int &n);  
void EnLepesem(int k1, int k2);  
void TeLepesed(int &k1, int &k2);
```

### Gyakorlás

A `minta.zip` fájlban letölthető egy egyszerű `Ellenfel` modul. A standard bemenet első sorára két egész számot tartalmazzon, a két kupacban lévő kavicsok számát.

### Korlátozások

Időlimit: 1 mp.

Memórialimit: 64MB

A programod nem írhat és nem olvashat semmilyen állományt, a standard outputra sem írhat!

A tesztek 30%-ában  $n, m \leq 500$ , további 30%-ában  $n, m \leq 3000$ .