

## Biztos fogadás

A szerencse alapvető része a fogadásoknak. Néhányan azzal növelik az esélyüket és a nyereségüket, hogy nagyobb ismeretre tesznek szert fogadásuk tárgyáról. Mi azonban más megközelítést alkalmazunk.

Különböző fogadóirodák különböző odd-okat ajánlanak ugyanahhoz a kimenethez. Egy  $x$  értékű odd azt jelenti, hogy ha 1 euróval fogadunk és helyesen jósoljuk meg a kimenetet, akkor  $x$  eurót kapunk vissza. Ha rosszul jósoljuk meg a kimenetet, akkor természetesen semmit sem kapunk vissza.

A fogadásban szereplő eseménynek két kimenete lehet.  $N$  fogadóiroda különböző odd-okat ajánl a két kimenetre. A felajánlott odd-ok bármelyik részalmazára lehet fogadni. Egy fogadóiroda mindkét kimenetére lehet akár fogadni, viszont ugyanazon fogadóiroda ugyanazon kimenetére egyenél többször nem. Minden fogadáshoz pontosan 1 eurót kell fizetni.

Az első kimenet bekövetkezése esetén  $a_i$  eurót kapunk vissza minden olyan  $i$  fogadóirodától, akinél erre a kimenetre fogadtunk. Hasonlóan, a második kimenet bekövetkezése esetén  $b_i$  eurót kapunk vissza minden olyan  $i$  fogadóirodától, akinél erre a kimenetre fogadtunk. Természetesen mindkét esetben 1 eurót kellett befizetnünk mindegyik fogadáshoz.

Készíts programot, amely meghatározza azt a garantált nyereséget az esemény kimenetelétől függetlenül, amelyet optimális fogadásokkal elérhetünk!

### Bemenet

A *standard bemenet* első sorában a fogadóirodák  $N$  száma ( $1 \leq N \leq 100000$ ) van. A további  $N$  sor mindegyike az  $i$ -edik fogadóiroda első és második kimenetére adott  $a_i$  és  $b_i$  odd értéket ( $1.0 \leq a_i, b_i \leq 1000.0$ ) tartalmazza. Az odd értékek legfeljebb 4 tizedesjegyet tartalmaznak.

### Kimenet

A *standard kimenet* egyetlen sorába a maximálisan garantált nyereséget kell kiírni 4 tizedesjegyre kerekítve.

Lebegőpontos  $x$  szám használata esetén a következőképpen lehet a fentieknek megfelelően kiírni az egyes programozási nyelvekben:

- C és C++: `printf("%.4lf", (double)x);`
- Java: `System.out.printf("%.4lf", x);`
- Pascal: `writeln(x:0:4);`
- Python 3: `print("%.4lf"%x)`
- C#: `Console.WriteLine(String.Format("0:0.0000", x));`

### Példa

Bemenet	Kimenet	Magyarázat
4 1.4 3.7 1.2 2 1.6 1.4 1.9 1.5	0.5000	Az optimális fogadási stratégia szerint az első fogadóirodánál a második kimenetre, a harmadik és a negyedik fogadóirodánál pedig az első kimenetre kell fogadni. Az első kimenet bekövetkezése esetén $1.6 + 1.9 - 3 = 0.5$ eurót keresünk, a második kimenet bekövetkezésekor $3.7 - 3 = 0.5$ eurót. Azaz 0.5 euró a garantált nyereség függetlenül az esemény kimenetelétől.

## **Korlátok**

Időlimit: 0,6 mp.

Memórialimit: 64 MB