



Zanesljiva stava

Omejitev časa: 2 s Omejitev pomnilnika: 128 MB

Pri stavah je sreča ena od bistvenih stvari. Nekateri ljudje izboljšajo svoje možnosti in zaslužek z dobrim poznavanjem tega, na kar stavijo. Mi pa bomo ubrali drugačno pot.

Različni pobiralci stav ponujajo za isti izid različna *razmerja*. (*Razmerje* x pomeni, da če staviš 1 evro in pravilno napoveš izid, dobiš x evrov. Če izid napoveš narobe, ne dobiš ničesar, vložek pa seveda ne glede na izid v vsakem primeru plačas.) Kaj če lahko zvito skleneš več stav tako, da boš zagotovo prišel do dobička? Seveda si želiš, da bi bil ta zanesljivi dobiček čim višji.

Dogodek, na katerega hočemo staviti, ima dva možna izida. Imamo n pobiralcev stav, ki ponujajo različna razmerja. Označimo z a_i razmerje, ki ga ponuja i -ti pobiralec na prvi izid, z b_i pa razmerje, ki ga ta pobiralec ponuja na drugi izid. Skleneš lahko stave na poljubno kombinacijo tako ponujenih razmerij, lahko celo pri istem pobiralcu staviš na oba izida. Vendar pa mora biti vsaka od tvojih stav za natanko 1 evro in pri posameznem pobiralcu ne moreš po večkrat staviti na isti izid.

Če pride do prvega izida, boš od vsakega pobiralca i , pri katerem si stavil na ta izid, dobil a_i evrov. Podobno, če pride do drugega izida, boš dobil b_i evrov od vsakega pobiralca, pri katerem si stavil na drugi izid. Seveda si v obeh primerih že plačal po 1 evro za vsako stavo, ki si jo sklenil.

Kolikšen je največji *zagotovljeni* dobiček (torej neodvisen od izida), če svoje stave skleneš optimalno?

Vhod

V prvi vrstici je število pobiralcev stav, n . Sledi n vrstic, od katerih i -ta vsebuje realni števili a_i in b_i , ločeni s presledkom — to sta razmerji, ki ju i -ti pobiralec ponuja za stave na prvi oz. drugi izid. Razmerja bodo podana na največ 4 decimalke.

Omejitve

- $1.0 \leq a_i, b_i \leq 1000.0$
- $1 \leq n \leq 100\,000$

Podnaloga 1 (20 točk)

- $n \leq 10$

Podnaloga 2 (40 točk)

- $n \leq 1\,000$

Podnaloga 3 (40 točk)

- brez dodatnih omejitev



Izhod

Izpiši največji zagotovljeni dobiček zaokrožen na natanko 4 decimalke.

Decimalna števila (floating point) lahko izpišeš s sledečimi ukazi:

- C and 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));`

Primer

Vhod

4
1.4 3.7
1.2 2
1.6 1.4
1.9 1.5

Izhod

0.5000

Comment

Najboljša strategija pri tem primeru je, da pri prvem pobiralcu stavimo na drugi izid, pri tretjem in četrtem pobiralcu pa na prvi izid. Če se zgodi prvi izid, bomo zaslužili $1,6 + 1,9 - 3 = 0,5$, pri drugem izidu pa $3,7 - 3 = 0,7$. Tako imamo torej zagotovljen dobiček 0,5 evrov ne glede na izid.