

Problema Riga Crypto

Fișier de intrare `stdin`
Fișier de ieșire `stdout`

Banii ușori se fac mulți, banii grei se fac puțini.

— Evil 'Claus

Cu banii proaspăt primiți de la olimpiade, Alex a decis să investească în criptomonede. Inițial, el deține 1 BTC și dorește să își crească această sumă. Pentru a realiza acest lucru, el va efectua schimburi pe parcursul următoarelor N zile.

Pe platforma pe care o folosește Alex, există K criptomonede, numerotate de la 1 la K , unde 1 reprezintă BTC.

În fiecare zi, Alex primește o listă de schimburi disponibile sub forma unei matrici A , unde $A[i][j]$ înseamnă că poate schimba o sumă ce o detine x , nu neaparat întreaga, din moneda i în $A[i][j] \cdot x$ din moneda j . Fiecare tranzacție durează exact o zi pentru a fi procesată, astfel încât Alex nu poate folosi banii schimbați în aceeași zi. Alex poate alege în cadrul unei zile să nu efectueze niciun schimb.

Cerință

După cele N zile, Alex dorește să rămână cu cât mai mult BTC. Determinați suma maximă de BTC cu care poate rămâne. Atenție, deoarece această valoare poate să nu fie un întreg, trebuie să afișați rezultatul cu o eroare maximă de 10^{-6} (vezi secțiunea de output pentru detalii suplimentare și cod).

Date de intrare

Pe prima linie se vor afla N și K , numărul de zile, respectiv numărul de criptomonede. Vor urma N matrici de K linii a câte K coloane. Pentru o matrice, numărul de pe linia i și coloana j reprezintă cursul în aceea zi pentru schimburile din i în j . Matricile vor conține numere reale $10^{-5} < A[i][j] < 10^5$.

Date de ieșire

Afișați numărul real x , care reprezintă suma maximă de BTC cu care poate rămâne Alex. Deoarece x este un număr real, răspunsul vostru trebuie să aibă o eroare relativă de cel mult 10^{-6} .

Se garantează că pentru răspunsul final: $x \leq 10^{13}$.

Fie x_{real} răspunsul corect și x_{conc} răspunsul concurentului. Se vor acorda puncte dacă:

$$\left| \frac{x_{\text{conc}}}{x_{\text{real}}} - 1 \right| \leq 10^{-6}$$

Pentru a asigura precizia calculelor, vă sugerăm să folosiți tipul de date `double` sau `long double` și să includeți următoarele în programul vostru:

```
#include <iomanip>
#include <iostream>

...

std::cout << std::fixed << std::setprecision(10) << x;
```

Restricții

- $1 \leq N \leq 80$
- $1 \leq K \leq 50$
- $10^{-5} < A[i][j] < 10^5$ pentru orice i și j .

#	Punctaj	Restricții
1	15	$N = 1$
2	15	$K = 2$
3	20	$N, K \leq 8$
4	50	Fără restricții suplimentare

Exemple

stdin	stdout
1 3 2.0 1.0 2.5 1.4 1.0 3.14159265359 3.0 3.0 3.0	2.0000000000

Explicații

$N = 1$ deci avem o singură zi de schimburi. Vom denumi moneda 2 “ETH” și moneda 3 “TON” pentru a face explicațiile mai ușoare. Schimburile posibile aici sunt:

- “BTC” -> $2.0 \cdot$ “BTC” (nu vom comenta asupra eticii investițiilor lui Alex)
- “BTC” -> $1.0 \cdot$ “ETH”
- “BTC” -> $2.5 \cdot$ “TON”
- “ETH” -> $1.4 \cdot$ “BTC”
- “ETH” -> $1.0 \cdot$ “ETH”
- “ETH” -> $3.14159265359 \cdot$ “TON”
- “TON” -> $3.0 \cdot$ “BTC”
- “TON” -> $3.0 \cdot$ “ETH”
- “TON” -> $3.0 \cdot$ “TON”

Atenție!, O tranzacție ia 24h până se efectuează, deci Alex nu poate folosi 2 tranzacții în aceeași zi. Deci NU poate “BTC” -> “ETH” -> “BTC” cu prețurile dintr-o singură zi. Totuși, poate face “BTC” -> “ETH” într-o zi și “ETH” -> “BTC” într-o zi ulterioară. In acest exemplu solutia optima e sa ia “BTC” -> $2 \cdot$ “BTC”.