

Mua hoa

Thành phố mà Nam sống có thể được mô phỏng bởi một lưới ô vuông kích thước $R * C$ được chia làm R hàng và C cột. Các hàng được đánh số từ 1 tới R từ trên xuống dưới, các cột được đánh số từ 1 tới C từ trái qua phải. Ô nằm trên hàng i và cột j được ký hiệu là ô (i, j) .

Nhà Nam ở ô $(1, 1)$ và nhà bạn gái Nam ở ô (R, C) .

Nam đang lên kế hoạch đến thăm nhà bạn gái mình một cách bất ngờ và tặng hoa cho cô ấy. Nam sẽ luôn đi theo đường đi ngắn nhất để đến nhà bạn gái, nói cách khác tại mỗi bước Nam luôn đi đến ô ngay bên phải hoặc ô ngay bên dưới ô hiện tại.

Tại mỗi ô, trừ ô $(1, 1)$ và (R, C) , đều có một người bán hoa. Người ở ô (i, j) bán $a(i, j)$ bông hoa hồng và $b(i, j)$ bông hoa tulip. Mỗi khi đi vào một ô, Nam sẽ mua tất cả hoa hồng và hoa tulip mà người ở ô đó bán. Nam phải mất $c(i, j)$ đồng để mua hoa ở ô (i, j) .

Yêu cầu: Nam chỉ có kinh phí là K đồng để mua hoa, do đó Nam sẽ không thể đi theo những đường đi mà tổng số tiền mua hoa vượt quá K . Gọi A, B lần lượt là tổng số bông hoa hồng, hoa tulip mà Nam mua được trên đường đi từ nhà mình đến nhà bạn gái. Nam biết bạn gái mình thích hoa hồng hơn hoa tulip nên Nam muốn tỉ lệ A/B là lớn nhất có thể. Hãy xác định giá trị lớn nhất đó.

Input: đọc từ file **proposal.in**

- Dòng đầu tiên chứa 3 số nguyên dương R, C, K ($4 \leq R * C \leq 300, K \leq 10^9$).
- R dòng sau, dòng thứ i ($1 \leq i \leq R$) chứa C số nguyên $a(i, 1), a(i, 2), \dots, a(i, C)$.
- R dòng sau, dòng thứ i ($1 \leq i \leq R$) chứa C số nguyên $b(i, 1), b(i, 2), \dots, b(i, C)$.
- R dòng sau, dòng thứ i ($1 \leq i \leq R$) chứa C số nguyên $c(i, 1), c(i, 2), \dots, c(i, C)$.
- Dữ liệu đảm bảo $a(1, 1) = a(R, C) = 0, b(1, 1) = b(R, C) = 0, c(1, 1) = c(R, C) = 0$ và với mọi ô (i, j) khác ô $(1, 1)$ và $(R, C), 1 \leq a(i, j), b(i, j), c(i, j) \leq 10^6$.

Output: ghi ra file **proposal.out**

Nếu mọi đường đi đến nhà bạn gái đều có số tiền mua hoa vượt quá K , in ra -1. Ngược lại in ra giá trị lớn nhất có thể của A / B làm tròn tới **đúng** 6 chữ số sau dấu phẩy.

Ví dụ:

proposal.in	proposal.out
2 2 100 0 2 3 0 0 3 5 0 0 70 80 0	0.666667
2 2 100 0 2	0.600000

3 0 0 3 5 0 0 170 100 0	
3 3 98 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 33 33 33 33 33 33 33 0	-1
2 3 15 0 1 3 1 1 0 0 1 2 6 1 0 0 1 18 9 1 0	1.000000