



The 5th Mohaghegh  
Collegiate Programming Contest  
May-10-2018



پنجمین دوره مسابقات  
برنامه نویسی **MCPC**

۲۰ اردیبهشت ۹۷

تعداد سوالات: ۸



## به خاطر یک تیشرت ۲ ایکس لارج

برای پنجمین دوره مسابقه MCPC تیشرت هایی توسط انجمن علمی گروه علوم کامپیوتر سفارش داده شده است. همه تیشرت ها سایز لارج دارند و تنها یک تیشرت ۲ ایکس لارج (2XL) بنفش رنگ موجود است. سینا دوکوهکی و محسن عبادپور برای بدست آوردن این تیشرت رقابت سختی دارند. این دو نفر بعد از صحبت های بسیار با یکدیگر به این نتیجه می رسند که هیچ راه حل مصالحت آمیزی برای این مشکل نمی توانند پیدا کنند. لذا هر دو تصمیم می گیرند به سبک گشتی ژاپنی سومو با سرعت به سمت یکدیگر بدونند و با یکدیگر برخورد کنند. کسی که از این برخورد جان سالم به در ببرد صاحب تیشرت ۲ ایکس لارج خواهد بود. سینا با تمام سرعت از سمت درب پارکینگ دانشگاه به سمت دانشکده فنی میدود و محسن نیز در جهت عکس از فنی به سمت درب پارکینگ حرکت می کند. سایر شرکت کنندگان مسابقه MCPC به دلیل جذابیت این مبارزه می خواهند در مکانی نزدیک به محل برخورد این دو نفر تجمع کنند تا در لحظه برخورد، عکس سلفی بگیرند. با نوشتن برنامه ای، به سایر شرکت کنندگان کمک نمایید تا محل برخورد را بصورت دقیق پیدا کرده و در آن مکان حضور یابند.

(فرض کنید مسیر بین فنی تا درب پارکینگ مسیری کاملاً مستقیم است و دو نفر کل مسیر را با سرعت ثابت میدوند)

ورودی

ورودی شامل سه عدد اعشاری می باشد. عدد اول فاصله میان فنی تا درب پارکینگ، عدد دوم سرعت حرکت محسن و عدد سوم سرعت سینا.

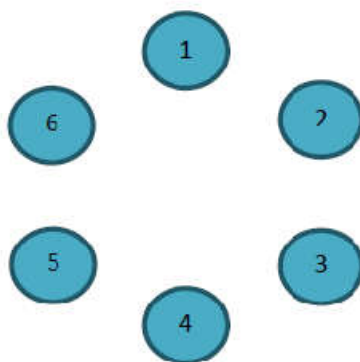
خروجی

فاصله دانشکده فنی تا محل برخورد

نمونه ورودی	نمونه خروجی
1200 4 8	400

## Ring

N people are playing a game. They sit in a circle with N seats. The following example shows a circle with 6 seats.



Starting from the person sitting on seat “1”, they count off clockwise by three. The person who has “three” loses and leaves the circle. Then, the rest continue playing the game by starting from the person who is the clockwise next to the leaving one until there is only one left. The one left is the winner.

In the circle above, the first person to be eliminated is person seating at seat “3”, then “4”, until only person seating at seat “1” is left.

Input

Single integer  $1 < N < 100000$  that indicates the number game participants.

Output

Single integer: the winner’s seat number

-----

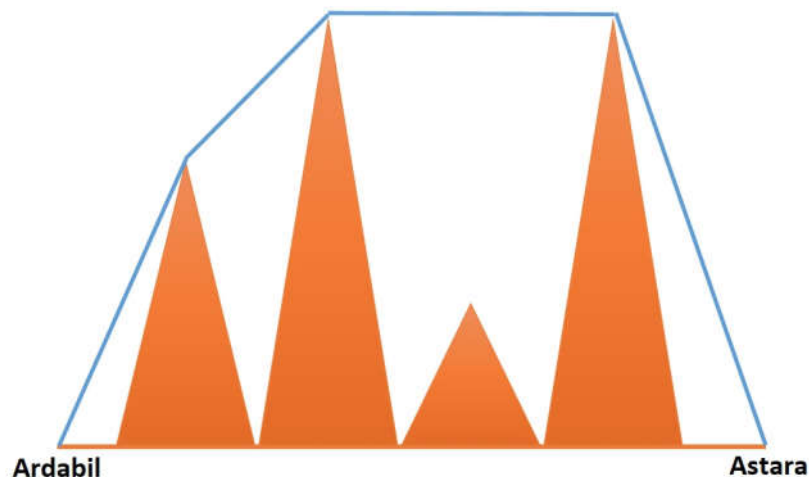
Sample input: 6

Sample output: 1



## کابل کشی

بهنام یوسفی مهر دبیر انجمن علمی به شدت دلش برای بهترین دوستش در آستارا تنگ شده است و به دلیل اینکه سرعت اینترنت بسیار کند است، می خواهد یک خط ارتباطی ویژه با سرعت بالا از اردبیل تا آستارا ایجاد کند. در مسیر بین اردبیل تا آستارا کوهستانی قرار گرفته که کابل های ارتباطی باید از روی آن رد شود. به او کمک کنید تا طول کابل را به حداقل برساند. برای هر قله در این مسیر، تیر برق برای عبور کابل بالای آن قله قرار دارد که کابل بایستی بین آنها متصل شود. یک مثال در شکل زیر نشان داده شده است.



اردبیل در موقعیت 0 قرار دارد و هر یک از N قله در موقعیت های 1 تا N واقع شده اند و آستارا در موقعیت N+1 قرار دارد. ارتفاع اردبیل و آستارا برابر 0 می باشد.

### ورودی

هر تست کیس با یک عدد N شروع می شود که نشان دهنده تعداد قله ها می باشد.  
سپس در خط بعدی N عدد صحیح وارد می شود که هر یک نشان دهنده ارتفاع قله ها می باشد.

### خروجی

برای هر تست کیس، طول حداقل کابل را محاسبه کنید. ( جواب را تا ۷ رقم اعشار گرد کرده و چاپ کنید)

Sample Input	Sample Output
4 3 4 1 4	10.6995968
5 0 0 4 0 0	10.0000000

## Balanced Diet

Every day, Amin buys one sweet from the candy store and eats it. The store has  $m$  types of sweets, numbered from 1 to  $m$ . Amin knows that a balanced diet is important and is applying this concept to his sweet purchasing. To each sweet type  $i$ , he has assigned a target fraction, which is a real number  $f_i$  ( $0 < f_i \leq 1$ ). He wants the fraction of sweets of type  $i$  among all sweets he has eaten to be roughly equal to  $f_i$ . To more precise, let  $s_i$  denote the number of sweets of type  $i$  that Amin has eaten, and let  $n = \sum_{i=0}^m s_i$ . We say the set of sweets is balanced if for every  $i$  we have

$$nf_i - 1 < s_i < nf_i + 1.$$

Amin has been buying and eating sweets for a while and during this entire time the set of sweets has been balanced. He is now wondering how many more sweets he can buy while still fulfilling this condition. Given the target fractions  $f_i$  and the sequence of sweets he has eaten so far, determine how many more sweets he can buy and eat so that at any time the set of sweets is balanced.

### Input

The input consists of three lines. The first line contains two integers  $m$  ( $1 \leq m \leq 10^5$ ), which is the number of types of sweets, and  $k$  ( $0 \leq k \leq 10^5$ ), which is the number of sweets Amin has already eaten. The second line contains  $m$  positive integers  $a_1, \dots, a_m$ . These numbers are proportional to  $f_1, \dots, f_m$ , that is,

$$f_i = \frac{a_i}{\sum_{j=1}^m a_j}.$$

It is guaranteed that the sum of all  $a_j$  is no larger than  $10^5$ . The third line contains  $k$  integers  $b_1, \dots, b_k$  ( $1 \leq b_i \leq m$ ), where each  $b_i$  denotes the type of sweet Amin bought and ate on the  $i^{\text{th}}$  day. It is guaranteed that every prefix of this sequence (including the whole sequence) is balanced.

### Output

Display the maximum number of additional sweets that Amin can buy and eat while keeping his diet continuously balanced. If there is no upper limit on the number of sweets, display the word forever.

#### Sample Input 1

```
6 5
2 1 6 3 5 3
1 2 5 3 5
```

#### Sample Output 1

```
1
```

#### Sample Input 2

```
6 1
2 1 6 3 5 3
1 2 5 3
```

#### Sample Output 2

```
forever
```



## Matrix Chain Product

If we are given a sequence of matrices to multiply,  $A_1, A_2, A_3, \dots, A_n$ , we can multiply them in any order, since matrix multiplication is associative, to generate the product.

If we have three matrices  $A_1, A_2$  and  $A_3$ . The product can be generated in two ways,  $A_1(A_2A_3)$  or  $(A_1A_2)A_3$ . The cost of multiplying a  $(n \times m)$  by an  $(m \times p)$  matrix is  $(n \times m \times p)$

So the choice of parenthesis could affect the cost of multiplying a sequence of matrices.

For example, if we have the matrices  $A_1$ , a  $5 \times 50$  matrix,  $A_2$ , a  $50 \times 10$  matrix and  $A_3$ , a  $10 \times 8$  matrix, the cost of  $A_1(A_2A_3)$  would be:  $(A_2A_3) 50 \times 10 \times 8 = 4000$ , a resulting  $50 \times 8$  matrix

$A_1(A_2A_3) 5 \times 50 \times 8 = 2000$ , a resulting  $5 \times 8$  matrix

Total Cost =  $4000 + 2000 = 6000$

the cost of  $(A_1A_2)A_3$  would be:  $(A_1A_2) 5 \times 50 \times 10 = 2500$ , a resulting  $5 \times 10$  matrix

$(A_1A_2)A_3 5 \times 10 \times 8 = 400$ , a resulting  $5 \times 8$  matrix

Total Cost =  $2500 + 400 = 2900$

input

Each line will contain a matrix chain. Each chain is represented by a series of positive integers,  $p, 0 < p \leq 100$ , terminated by a 0. For example the chain above would be "5 50 10 8 0". Note that  $N$  matrices are represented by  $N + 1$  non-zero numbers.  $2 \leq N \leq 10$ .

output

Each line will contain the lowest cost of multiplying the matrix chain.

The highest cost will not exceed 2000000000.

Input	Output
5 50 10 8 0	2900
4 12 6 9 12 0	936
25 30 10 5 0	5250



## Rasterizing Triangles

In computer graphics, a common task is to rasterize an image, converting geometrical shapes into a 2D array of pixels to be rendered on the screen. In this problem, we will rasterize randomly selected pixels of an image consisting only of a single triangle. In order to do this correctly, we must determine whether each pixel is inside the triangle and should get the triangle's color, or if it's outside the triangle and should receive the background color, whatever that is.

Hint: Don't assume that the triangle's vertices are sorted in any way in the input

### Input

Input consists of a triangle (described by three 2D vertices) and the coordinates of the pixel. For simplicity, each triangle and pixel coordinate is positive and a multiple of 0.5 (i.e. 0.0, 0.5, 1.0, 1.5, etc.).

First line: six, space delimited, coordinates describing the 3 vertices of the triangle (i.e.  $x_1 y_1 x_2 y_2 x_3 y_3$ )

Second line: a pixel vertex, as two, space delimited, coordinates ( $p_x p_y$ )

### Output

Output is 0 if the pixel is outside of the triangle and 1 if it's inside

Sample Input	Sample Output
1.0 1.0 2.0 2.0 3.0 1.0 2.0 1.5	1
1.0 1.0 2.0 2.0 3.0 1.0 0.5 1.5	0



## کانال ارتباطی مخفی

یک شرکت از اینترنت عمومی برای ارسال تماس های تلفنی خود استفاده می کند. داده های صدا بایستی قبل از ارسال روی اینترنت رمزگذاری شوند. الگوریتم رمزگذاری بصورت زیر است:

۱. به هر رقم عدد ارسالی ۵ واحد اضافه کن و نتیجه جمع هر رقم را بر ۱۰ تقسیم کن و باقیمانده را بجای رقم اولیه قرار بده
۲. جای اولین رقم را با رقم چهارم عوض کن
۳. جای دومین رقم و سومین رقم را با یکدیگر عوض کن

## ورودی

یک عدد صحیح مثبت چهار رقمی

## خروجی

یک عدد صحیح چهار رقمی رمزگذاری شده

Sample Input	Sample Output
0123	8765
5890	5430
9999	4444





## Take a Walk

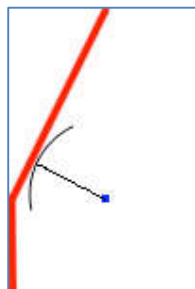
Like any other 10-year-old, Little Billy loves to eat candy. However, every day after school, his parents give him a specific path he has to take home to make sure he doesn't go to the candy store. Billy is given a number of points, starting at his school and ending at home, and he walks directly from the 1<sup>st</sup>, to the 2<sup>nd</sup>, then to the 3<sup>rd</sup>, and so on. Even though he isn't allowed to go to the store, Little Billy wants to know the closest distance he gets to the store on his way home. Being his only imaginary friend who knows how to program, help him by computing this distance.

### Input

The first line of each set contains a number  $1 \leq N \leq 10$ , the number of points on Little Billy's walk back home. The next  $N$  lines each contain two numbers separated by a single space,  $-1000 \leq X, Y \leq 1000$ , the coordinates of points for the walk. The last line of each set contains two numbers separated by a single space,  $-1000 \leq C_x, C_y \leq 1000$ , the location of the forbidden candy store.

### Output

The closest distance to the candy store, from anywhere along the walk. Round to 2 decimal places. Print trailing zeros. E.g. *1.00*



Sample Input	Sample Output
3 0 0 0 1 1 3 1 1	0.89