



## Łamana

Azerbejdżan słynie z dywanów. Jako mistrz projektujący wzory zamierzasz stworzyć nowy wzór będący **łamaną**. Łamana to ciąg  $t$  odcinków na płaszczyźnie zdefiniowany za pomocą  $t + 1$  punktów  $p_0, \dots, p_t$  w następujący sposób. Dla każdego  $0 \leq j \leq t - 1$ , kolejny odcinek łamanej łączy punkty  $p_j$  oraz  $p_{j+1}$ .

Aby stworzyć nowy wzór, naniosłeś już  $n$  **kropek** na dwuwymiarowej płaszczyźnie. Współrzędne kropki  $i$  ( $1 \leq i \leq n$ ), to  $(x[i], y[i])$ . **Żadne dwa punkty nie mają takiej samej współrzędnej  $x$  lub  $y$ .**

Chcesz znaleźć ciąg punktów  $(sx[0], sy[0]), (sx[1], sy[1]), \dots, (sx[k], sy[k])$ , określających łamaną, która:

- zaczyna się w  $(0, 0)$  (tzn.  $sx[0] = 0$  i  $sy[0] = 0$ ),
- zawiera wszystkie kropki (niekoniecznie jako końce odcinków),
- składa się jedynie z poziomych bądź pionowych odcinków (dwa kolejne punkty łamanej mają tę samą współrzędną  $x$  lub  $y$ ).

Łamana może się ze sobą przecinać lub nawet nachodzić na siebie w dowolny sposób. Formalnie: każdy punkt płaszczyzny może należeć do dowolnej liczby odcinków łamanej.

Jest to zadanie typu „output only” z częściową punktacją. Masz dane 10 plików wejściowych określających położenie kropek.

Dla każdego z plików wejściowych należy wygenerować plik wyjściowy definiujący łamaną zgodną z wymaganiami zadania. Dla każdego z plików wyjściowych opisujących poprawne łamane Twój wynik będzie zależał od **liczby odcinków** tworzących łamaną (patrz tabela przeliczników w dalszej części).

Nie należy zgłaszać żadnego kodu dla tego zadania.

## Format wejścia

Każdy plik wejściowy ma następujący format

- wiersz 1:  $n$
- wiersze  $1 + i$  (dla  $1 \leq i \leq n$ ):  $x[i] \ y[i]$

## Format wyjścia

Każdy plik wyjściowy musi mieć następujący format:

- wiersz 1:  $k$
- wiersze  $1 + j$  (dla  $1 \leq j \leq k$ ):  $sx[j] \ sy[j]$

Zwróć uwagę na to, że drugi wiersz powinien zawierać  $sx[1]$  i  $sy[1]$  (tj. wyjście **nie powinno** zawierać  $sx[0]$  i  $sy[0]$ ). Każda z liczb  $sx[i]$  oraz  $sy[i]$  powinna być całkowita.

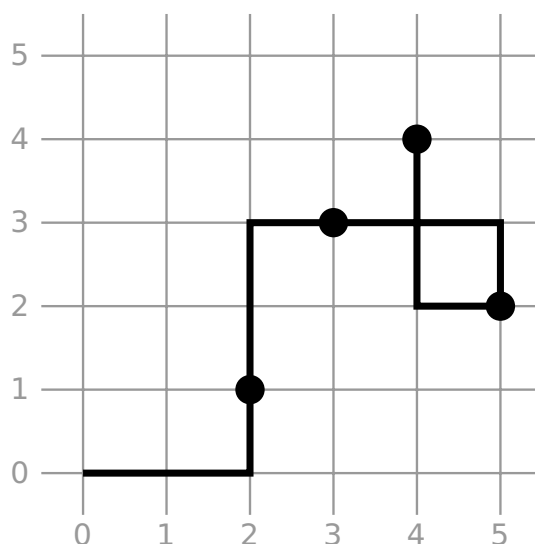
## Przykład

Dla przykładowego wejścia

```
4
2 1
3 3
4 4
5 2
```

jednym z poprawnych wyjść jest:

```
6
2 0
2 3
5 3
5 2
4 2
4 4
```



Uwaga: powyższego przykładu nie ma dziś wśród plików wejściowych.

## Ograniczenia

- $1 \leq n \leq 100\,000$
- $1 \leq x[i], y[i] \leq 10^9$
- Wszystkie wartości  $x[i]$  oraz  $y[i]$  są całkowite.
- Żadne dwie kropki nie mają tych samych współrzędnych  $x$  ani  $y$ , czyli  $x[i_1] \neq x[i_2]$  oraz  $y[i_1] \neq y[i_2]$  dla  $i_1 \neq i_2$ .
- $-2 \cdot 10^9 \leq sx[j], sy[j] \leq 2 \cdot 10^9$
- Żaden plik zgłoszony do oceny (zzipowany czy nie) nie może mieć rozmiaru przekraczającego 15MB.

## Ocena rozwiązania

Za każdy z testów dostaniesz co najwyżej 10 punktów. Twój plik wyjściowy otrzyma 0 punktów, jeśli nie definiuje łamanej o wymaganych własnościach. W przeciwnym razie wynik zostanie ustalony w odniesieniu do skali zależnej od malejącego ciągu  $c_1, \dots, c_{10}$ , określonej dla każdego testu z osobna.

Przyjmijmy, że Twoja łamana składa się z  $k$  odcinków. Otrzymasz wtedy

- $i$  punktów, jeśli  $k = c_i$  (dla  $1 \leq i \leq 10$ ),
- $i + \frac{c_i - k}{c_i - c_{i+1}}$  punktów, jeśli  $c_{i+1} < k < c_i$  (dla  $1 \leq i \leq 9$ ),
- 0 punktów, jeśli  $k > c_1$ ,
- 10 punktów, jeśli  $k < c_{10}$ .

Ciągi wartości  $c_1, \dots, c_{10}$  dla kolejnych testów są podane poniżej.

Testcases	01	02	03	04	05	06	07-10
$n$	20	600	5 000	50 000	72 018	91 891	100 000
$c_1$	50	1 200	10 000	100 000	144 036	183 782	200 000
$c_2$	45	937	7 607	75 336	108 430	138 292	150 475
$c_3$	40	674	5 213	50 671	72 824	92 801	100 949
$c_4$	37	651	5 125	50 359	72 446	92 371	100 500
$c_5$	35	640	5 081	50 203	72 257	92 156	100 275
$c_6$	33	628	5 037	50 047	72 067	91 941	100 050
$c_7$	28	616	5 020	50 025	72 044	91 918	100 027
$c_8$	26	610	5 012	50 014	72 033	91 906	100 015
$c_9$	25	607	5 008	50 009	72 027	91 900	100 009
$c_{10}$	23	603	5 003	50 003	72 021	91 894	100 003

# Wizualizator

W paczce do tego zadania jest skrypt, który pozwoli Ci zwizualizować pliki wejściowe i wyjściowe.

Aby zwizualizować plik wejściowy użyj następującego wywołania:

```
python vis.py [input file]
```

Możesz również zwizualizować Twoje rozwiązanie dla dowolnego pliku wejściowego, używając poniższego wywołania. Ze względów technicznych wizualizator pokaże jedynie **pierwszych 1000 odcinków** pliku wyjściowego.

```
python vis.py [input file] --solution [output file]
```

Przykład takiego wywołania:

```
python vis.py examples/00.in --solution examples/00.out
```