



Línea quebrada

Azerbaijan es famoso por sus alfombras. Como maestro diseñador de alfombras, deseas crear un nuevo diseño dibujando una **línea quebrada**. Una línea quebrada es una secuencia de t segmentos en el plano bidimensional, definida por una secuencia de $t + 1$ puntos p_0, \dots, p_t de la siguiente manera. Para cada $0 \leq j \leq t - 1$ hay un segmento que conecta los puntos p_j y p_{j+1} .

Para crear el nuevo diseño, ya has marcado n **puntos** en el plano bidimensional. Las coordenadas del punto i ($1 \leq i \leq n$) son $(x[i], y[i])$. **No existen dos puntos con la misma coordenada x, ni con la misma coordenada y.**

Ahora quieres encontrar una secuencia de puntos $(sx[0], sy[0]), (sx[1], sy[1]), \dots, (sx[k], sy[k])$, que defina una línea quebrada tal que:

- comience en $(0, 0)$ (es decir, $sx[0] = 0$ y $sy[0] = 0$),
- contenga todos los puntos (no necesariamente como los extremos de los segmentos), y
- consista únicamente de segmentos horizontales o verticales (dos puntos consecutivos en la definición de la línea quebrada tienen igual coordenada x o igual coordenada y).

Se permite tanto que la línea quebrada se corte a sí misma, como que se superponga consigo misma. Formalmente, cada punto del plano puede pertenecer a cualquier cantidad de segmentos de la línea quebrada.

Este es un problema de tipo output-only, con puntaje parcial. Recibes 10 archivos de entrada, que especifican las ubicaciones de los puntos. Por cada archivo de entrada, debes enviar un archivo de salida que describa una línea quebrada con las propiedades requeridas. Por cada archivo de salida que describa una línea quebrada válida, el puntaje depende de la **cantidad de segmentos** en la línea quebrada (ver la puntuación abajo).

No debes enviar ningún código fuente para este problema.

Formato de entrada

Cada archivo de entrada tiene el siguiente formato:

- línea 1: n
- línea $1 + i$ (para $1 \leq i \leq n$): $x[i] \ y[i]$

Formato de salida

Cada archivo de salida debe tener el siguiente formato:

- línea 1: k
- línea $1 + j$ (para $1 \leq j \leq k$): $sx[j] \ sy[j]$

Notar que la segunda línea debe contener $sx[1]$ y $sy[1]$ (es decir, la salida **no debe** incluir los valores $sx[0]$ y $sy[0]$). Cada $sx[j]$ y $sy[j]$ debe ser un entero.

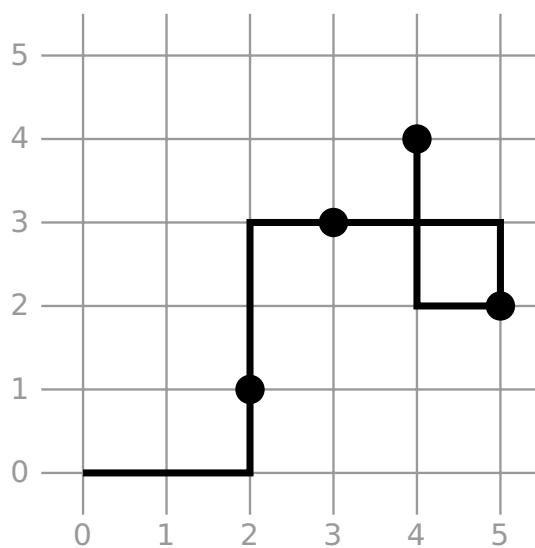
Ejemplo

Para la siguiente entrada de ejemplo:

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

una posible salida válida es:

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



Por favor, notar que este ejemplo no se encuentra entre los verdaderos archivos de entrada del problema.

Restricciones

- $1 \leq n \leq 100\,000$
- $1 \leq x[i], y[i] \leq 10^9$
- Todos los valores $x[i]$ y $y[i]$ son enteros.
- Ningún par de puntos tiene la misma coordenada x ni la misma coordenada y , es decir, $x[i_1] \neq x[i_2]$ y $y[i_1] \neq y[i_2]$ para $i_1 \neq i_2$.
- $-2 \cdot 10^9 \leq sx[j], sy[j] \leq 2 \cdot 10^9$
- El tamaño de cada archivo enviado (tanto si es un archivo de salida, como si es un archivo comprimido) no puede exceder 15MB.

Puntaje

Por cada caso de prueba, se pueden conseguir hasta 10 puntos. La salida para un caso de prueba obtiene 0 puntos si no especifica una línea quebrada con las propiedades requeridas. De lo contrario, el puntaje se determina utilizando una secuencia decreciente c_1, \dots, c_{10} , que varía según el caso de prueba.

Un archivo de salida que describe una línea quebrada válida con k segmentos obtiene:

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

La secuencia c_1, \dots, c_{10} correspondiente a cada caso de prueba se indica a continuación.

Casos	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

Visualizador

En los archivos adjuntos de este problema, se puede encontrar un script que permite visualizar los archivos de entrada y salida.

Para visualizar un archivo de entrada, utilizar el siguiente comando:

```
python vis.py [archivo de entrada]
```

Se puede también visualizar una solución para cierta entrada, utilizando el siguiente comando. Debido a limitaciones técnicas, el visualizador provisto muestra únicamente **los primeros 1000 segmentos** del archivo de salida.

```
python vis.py [archivo de entrada] --solution [archivo de salida]
```

Ejemplo:

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