



Broken Line

Azerbaijão é famoso por seus tapetes. Como um estilista de tapetes mestre você quer criar um novo estilo desenhando uma **linha quebrada**. Uma linha quebrada é uma sequência de t segmentos de linha em um plano bidimensional, o que é definido por uma sequência de $t + 1$ pontos p_0, \dots, p_t da seguinte maneira: para cada $0 \leq i \leq t - 1$ existe um segmento conectando os pontos p_j e p_{j+1} .

Para criar o novo estilo, você já tem n **pontos** marcados em um plano bidimensional. As coordenadas do ponto i ($1 \leq i \leq n$) são $(x[i], y[i])$. **Não há dois pontos com a mesma coordenada x ou com a mesma coordenada y.**

Você agora quer encontrar uma sequência de pontos $(sx[0], sy[0]), (sx[1], sy[1]), \dots, (sx[k], sy[k])$, que definem uma linha quebrada que

- começa em $(0, 0)$ (isto é, $sx[0] = 0$ e $sy[0] = 0$),
- contém todos os pontos (não necessariamente como pontos finais dos segmentos), e
- consiste unicamente de segmentos horizontais ou verticais (dois pontos consecutivos definindo a linha quebrada possuem uma mesma coordenada x ou uma mesma coordenada y).

É permitido que a linha quebrada se intersecte ou se sobreponha a si mesma de qualquer forma. Formalmente, cada ponto do plano pode pertencer a qualquer número de segmentos da linha quebrada (veja seção Pontuação abaixo).

Esta é uma tarefa apenas-saída com pontuação parcial. Você recebe 10 arquivos de entrada especificando as coordenadas dos pontos. Para cada arquivo de entrada, você deve submeter um arquivo de saída descrevendo uma linha quebrada com as propriedades requisitadas. Para cada arquivo de saída que descreve uma linha quebrada válida sua pontuação depende do **número de segmentos** da linha quebrada.

Você não deve submeter qualquer código fonte para esta tarefa.

Formato de entrada

Cada arquivo de entrada está no seguinte formato:

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

Formato de saída

Cada arquivo de saída deve estar no seguinte formato:

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

Note que a segunda linha deve conter $sx[1]$ e $sy[1]$ (i.e., a saída **não deve** conter $sx[0]$ e $sy[0]$). Cada $sx[j]$ e $sy[j]$ deve ser um inteiro.

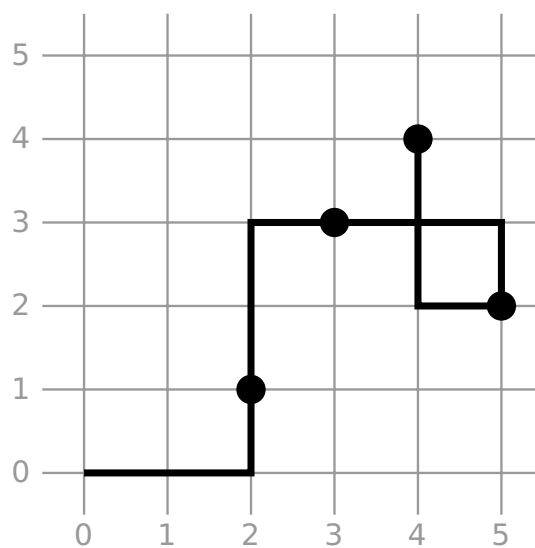
Exemplo

Para o exemplo de entrada:

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

uma possível saída válida é:

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



Por favor note que esse exemplo não está entre as entradas reais desta tarefa.

Restrições

- $1 \leq n \leq 100\,000$
- $1 \leq x[i], y[i] \leq 10^9$
- Todos os valores de $x[i]$ e $y[i]$ são inteiros.
- Dois pontos não possuem a mesma coordenada x ou a mesma coordenada y , i.e. $x[i_1] \neq x[i_2]$ e $y[i_1] \neq y[i_2]$ para $i_1 \neq i_2$.
- $-2 \cdot 10^9 \leq sx[j], sy[j] \leq 2 \cdot 10^9$
- O tamanho de cada arquivo submetido (seja um arquivo de saída ou um arquivo zipado) não pode exceder 15MB.

Pontuação

Para cada caso de teste, você pode receber até 10 pontos. Sua saída para um caso de teste irá receber 0 pontos se ela não especifica uma linha quebrada com as propriedades requisitadas. Caso contrário, a pontuação será determinada usando uma sequência decrescente c_1, \dots, c_{10} , que varia entre os casos de teste.

Assuma que sua solução é uma linha quebrada válida e consiste em k segmentos. Então, você receberá

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

A sequência c_1, \dots, c_{10} para cada caso de teste é dada abaixo.

Casos de teste	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

Nos pacotes anexos desta tarefa, há um script que permite visualizar arquivos de entrada e saída.

Para visualizar um arquivo de entrada, use o seguinte comando:

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

Você também pode visualizar sua solução para alguma entrada, usando o seguinte comando. Devido a limitações técnicas, o visualizador providenciado mostra apenas **os primeiros 1000 segmentos** do arquivo de saída.

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

Exemplo:

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