



Broken Line

O Azerbaijão é famoso pelos seus tapetes. Sendo um especialista no desenho de tapetes, queres fazer um novo design desenhando uma **linha partida**. Uma linha partida é uma sequência de t segmentos de reta num plano bi-dimensional, que é definido por uma sequência de $t + 1$ pontos p_0, \dots, p_t como a seguir definido. Para cada $0 \leq j \leq t - 1$ existe um segmento que liga os pontos p_j e p_{j+1} .

Para fazer o novo design, tens já marcadas n **pintas** (pontos marcados) no plano. As coordenadas da pinta i ($1 \leq i \leq n$) são $(x[i], y[i])$. **Nenhum par de pintas tem a mesma coordenada x ou a mesma coordenada y.**

Queres agora descobrir uma sequência de pontos $(sx[0], sy[0]), (sx[1], sy[1]), \dots, (sx[k], sy[k])$, que defina uma linha partida que:

- comece em $(0, 0)$ (isto é, $sx[0] = 0$ e $sy[0] = 0$),
- contenha todas as pintas (não necessariamente como extremidades dos segmentos), e
- consista somente em segmentos horizontais ou verticais (dois pontos consecutivos da linha partida devem tem uma coordenada x ou y igual).

É permitido que a linha partida se interseste ou se sobreponha a si própria de qualquer maneira. Formalmente, cada ponto do plano pode pertencer a qualquer número de segmentos da linha partida.

Este é um problema de "output-only" com pontuações parciais. São-te dados 10 ficheiros de input especificando as localizações das pintas. Para cada ficheiro de input, deves submeter um ficheiro de output descrevendo a linha partida com as propriedades atrás descritas. Para cada ficheiro de output que descreva uma linha partida válida, a tua pontuação depende do **número de segmentos** na linha partida (vê a secção Pontuação à frente).

Não é suposto submeteres qualquer código-fonte para este problema.

Formato de Input

Cada ficheiro de input está no seguinte formato:

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

Formato de output

Cada ficheiro de output deve estar no seguinte formato:

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

Nota que a segunda linha deve conter $sx[1]$ e $sy[1]$ (isto é, o output **não deve** conter $sx[0]$ e $sy[0]$). Cada $sx[j]$ e $sy[j]$ deve ser um inteiro.

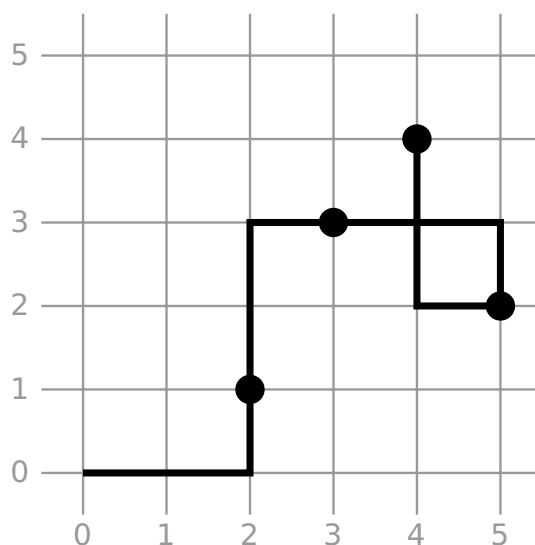
Exemplo

Para o input:

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

Um output válido possível é:

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



Por favor nota que o exemplo não está nos inputs reais que vão ser usados no problema.

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.
- Nenhum par de pintas tem a coordenada x ou a mesma coordenada y , ou seja, $x[i_1] \neq x[i_2]$ 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 ficheiro submetido (um output ou um ficheiro zip) não pode exceder 15MB.

Pontuação

Em cada caso de teste podes obter um máximo de 10 pontos. O teu output para um caso de teste obterá 0 pontos se não especificar uma linha partida com as propriedades desejadas. Caso contrário (ou seja, se a linha partida for válida), a pontuação será determinada usando a sequência decrescente c_1, \dots, c_{10} , que varia consoante o caso de teste.

Assumindo que a tua solução é uma linha partida válida consistindo em k segmentos, terás:

- 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 na tabela seguinte:

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 anexos deste problema, existe um script que permite visualizar os ficheiros de input e output.

Para visualizar um ficheiro de input, usa o seguinte comando:

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

Podes também visualizar a tua solução para um dado input usando o seguinte comando. Devido a limitações técnicas, o visualizador providenciado mostra apenas **os primeiros 1000 segmentos** do ficheiro de output.

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

Exemplo:

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