



СЫНЫК СЫЗЫК

Азербайжан өзүнүн килемдери менен белгилүү. Килем устасы катары сиз **сынык сызык** менен жаңы дизайнды жасагыңыз келет. **Сынык сызык** эки өлчөмдүү тегиздикте t кесиндинин бирикмеси. Ал $(t + 1)$ чоку аркылуу аныкталат. Ар катар кеткен p_j жана p_{j+1} эки чоку бир кесиндини билдирет. Ошондо сынык сызыктын чокулары p_0, \dots, p_t .

Жаңы дизайнды жасоо үчүн сиз n Точканы эки өлчөмдүү тегиздикте белгиледиңиз. i -инчи Точканын координаталары i ($1 \leq i \leq n$) булар $(x[i], y[i])$. **Эки Точканын x координаталары барабар эмес жана y координаталары барабар эмес.**

Сынык сызыктын $(sx[0], sy[0]), (sx[1], sy[1]), \dots, (sx[k], sy[k])$ чокуларын төмөндөгү шартка карап тапкыла:

- $(0, 0)$ менен башталат (башкача айтканда, $sx[0] = 0$ жана $sy[0] = 0$),
- Бардык Точканы камтыш керек,
- кесиндилер горизанталдуу же вертикалдуу болуш керек.

Сынык сызык өзүн өзү кесиши же өзүнүн үстүнөн өтүшү мүмкүн.

Бул суроо output-only . Канча аз сызык менен чыгарылса ошончо көп балл берилет.

Сизге Точкаларды жайгаштыруучу 10 input файл берилет. For each input file, you should submit an output file describing a broken line with the required properties. For each output file that describes a valid broken line your score depends on the **number of segments** in the broken line (see Scoring below).

You are not supposed to submit any source code for this task.

Input format

Each input file is in the following format:

- line 1: n
- line $1 + i$ (for $1 \leq i \leq n$): $x[i] \ y[i]$

Output format

Each output file must be in the following format:

- line 1: k
- line $1 + j$ (for $1 \leq j \leq k$): $sx[j] \ sy[j]$

Note that the second line should contain $sx[1]$ and $sy[1]$ (i.e., the output **should not** contain $sx[0]$ and $sy[0]$). Each $sx[j]$ and $sy[j]$ should be an integer.

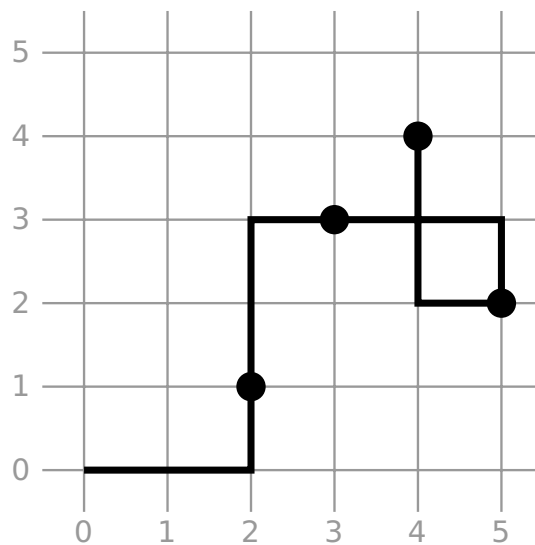
Example

For the sample input:

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

a possible valid output is:

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



Please note this example is not among the actual inputs of this task.

Constraints

- $1 \leq n \leq 100\,000$
- $1 \leq x[i], y[i] \leq 10^9$
- All values of $x[i]$ and $y[i]$ are integers.
- No two dots have the same x or the same y coordinates, i.e. $x[i_1] \neq x[i_2]$ **and** $y[i_1] \neq y[i_2]$ for $i_1 \neq i_2$.
- $-2 \cdot 10^9 \leq sx[j], sy[j] \leq 2 \cdot 10^9$
- The size of each submitted file (either an output or a zipped file) cannot exceed 15MB.

Scoring

For each test case, you can get up to 10 points. Your output for a test case will get 0 points if it does not specify a broken line with the required properties. Otherwise, the score will be determined using a decreasing sequence c_1, \dots, c_{10} , which varies by testcase.

Assume that your solution is a valid broken line consisting of k segments. Then, you will get

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

The sequence c_1, \dots, c_{10} for each testcase is given below.

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

Visualizer

In the attachments of this task, there is a script that allows you to visualize input and output files.

To visualize an input file, use the following command:

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

You can also visualize your solution for some input, using the following command. Due to technical limitations, the provided visualizer shows only **the first 1000 segments** of the output file.

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

Example:

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