French-Australian Regional Informatics Olympiad

Friday 9th March, 2018

Duration: 4 hours

3 questions

All questions should be attempted

Problem 1
Shadow Architecture

Input File: *standard input*
Output File: *standard output*

Time and Memory Limits: 1 second, 256 MB

You are an intern for an avant-garde architect renowned for their beautiful but highly unstable designs. Your current project is to relocate a mansion for a reclusive oligarch who dislikes afternoon sunlight.

The mansion to be relocated is on the eastern side of an infinitely long street. The architect has already built another mansion in a similar style on the western side of the street, which you will use for shade. Since the street is infinitely long, all positions are measured relative to a fixed point, the "end" of the street, though the street does span beyond that. Both mansions are N storeys tall, and every storey is both contiguous and has at least one metre of overlap with the storey below. Further suppose that the i-th story on the westerly mansion spans from s_i to e_i, then a shadow will be cast on the i-th storey of your client's mansion from s_i to e_i.

Money is no object to your client, and you may move his mansion to be anywhere along his infinitely long street. Your job now is to figure out the maximum area of your client's mansion that can be in shade.

Input

- The first line of input contains N, the number of storeys in each mansion.
- The following N lines contain two integers each. The i-th line of which is $a_i\ b_i$, which indicates that the i-th storey of the west side mansion begins a_i metres from the end of the street and ends $b_i + 1$ metres from the end of the street.
- The following N lines contain two integers each. The i-th line of which is $c_i\ d_i$, which indicates that the i-th storey of your client's mansion begins c_i metres from the end of the street and ends $d_i + 1$ metres from the end of the street.

Output

Output must consist of one integer, the maximum area in shade that can be achieved on your client's mansion.

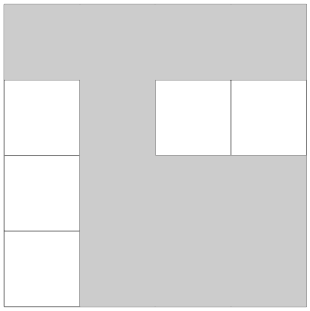
Sample Input

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

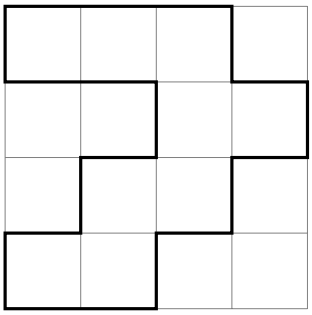
Sample Output

```
7
```

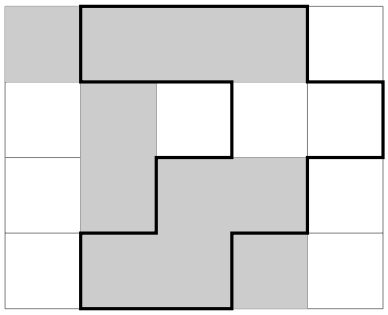
Explanation



West side mansion



Client mansion



Mansion shadowing

Above on the left we see the shape of the mansion on the west side of the street. This is represented in gray for the shadow it is casting. In the middle we see the shape of our client's mansion, this may be shifted arbitrarily to maximise the amount of shadow that falls on it.

By moving the client mansion one meter to the right, the entirety of its first, second and third floors are covered by afternoon shadow. This results in seven $(2 + 2 + 3)$ units of shade falling on the client's mansion. This is the maximum amount of shade, thus the correct output is 7.

Subtasks & Constraints

For all subtasks, $1 \leq N \leq 100\,000$, $0 \leq a_i \leq b_i \leq N$, $0 \leq c_i \leq d_i \leq N$.

- For Subtask 1 (30 points), $1 \leq N \leq 300$.
- For Subtask 2 (30 points), $1 \leq N \leq 3000$.
- For Subtask 3 (40 points), no additional constraints.

Problem 2
Greed

Input File: *standard input*
Output File: *standard output*

Time and Memory Limits: 3.5 seconds, 256 MB

Happy birthday! It is time to cut your cake! Including you, there are $K+1$ hungry people who are eager for some of your R row by C column rectangular cake. To ensure fairness, there are some rules about how you may cut your cake.

- You must make K cuts, in order to split the cake into $K+1$ pieces.
- Every cut must be made parallel to the sides of the rectangle.
- Every cut must begin on either the left or bottom side of the cake, and must continue until it hits either the opposite side of the cake or an existing cut.
- There are H allowed locations for horizontal cuts and V locations allowed for vertical cuts.
- Once all cuts have been made, you will receive the largest piece of cake.

You love cake, but do not wish to be seen to be greedy. In order to avoid this fate, you must make the area of the largest piece of cake as small as possible.

Input

- The first line of input contains five integers $R\,C\,K\,H\,V$.
- The following H lines contain one integer each, the i-th of which being y_i. This indicates a horizontal cut can be made y_i units up from the bottom side of the cake. These will be distinct and strictly increasing.
- The following V lines contain one integer each, the i-th of which being x_i. This indicates a vertical cut can be made x_i units to the right of the left side of the cake. These will be distinct and strictly increasing.

Output

Output must consist of a single integer representing the area of the smallest possible largest slice of cake.

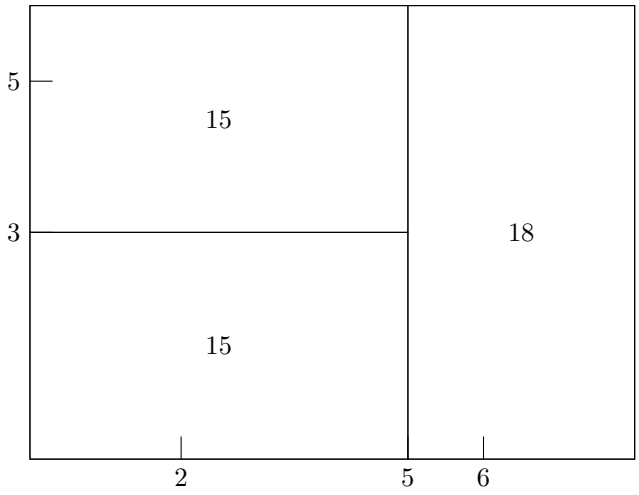
Sample Input

```
6 8 2 2 3
3
5
2
5
6
```

Sample Output

```
18
```

Explanation



In this example, it is possible for us to create vertical cuts at 2, 5, and 6, as well as horizontal cuts at 3 and 5.

By first cutting the cake vertically at position 5, and then horizontally at position 3, we can produce two slices of area 15 and once slice of area 18. This is the smallest we can make the largest slice, and so we output 18.

Subtasks & Constraints

For all subtasks, $1 \leq K \leq H + V$, $0 \leq H, V \leq 1500$, $2 \leq R, C$, $4 \leq RC \leq 2^{30}$, $1 \leq x_i < C$ and $1 \leq y_i < R$.

- For Subtask 1 (10 points), $H, V \leq 10$.
- For Subtask 2 (15 points), $K = H + V \leq 40$.
- For Subtask 3 (15 points), $H, V \leq 40$.
- For Subtask 4 (20 points), $H, V \leq 100$.
- For Subtask 5 (20 points), $K \leq 100$, $H, V \leq 300$.
- For Subtask 6 (20 points), no additional constraints.

Problem 3
Balancing Sculptures

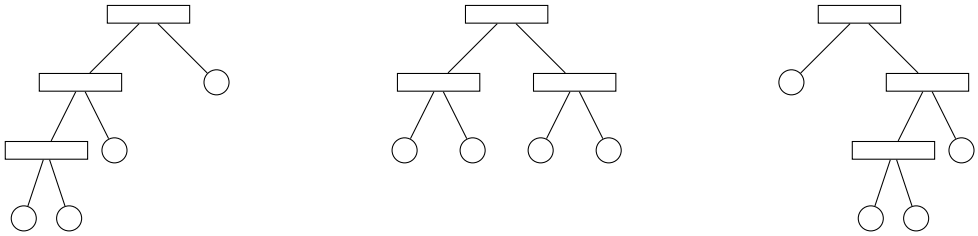
Input File: `xy.in`
Output File: `output_xy.txt`

Output Only Task

Congratulations on becoming the Chief Balance Officer at a hip new startup! Your primary responsibility is to produce sculptures which are as balanced as you can achieve.

Your latest client has tasked you with designing a sculpture that showcases a collection of circular wooden blocks, each with a particular weight. You will assemble them into a single sculpture using wire rods, each of negligible weight. You may use as many of these rods as you need.

Each rod has two hooks below it. To each hook, you **must** attach either a block or another rod – an empty hook is a plague to the eyes! All blocks must participate in the construction of the sculpture, and you must be able to pick up the entire sculpture from the topmost rod.



The *imbalance* of a rod is the absolute difference in weight between what is hanging from each of its two hooks. The *total imbalance* of the sculpture is the sum of imbalances over each of its rods.

Your client is quite the connoisseur of wooden blocks, and simply can't decide which of her collections of blocks to use. For each of them, find an arrangement of blocks and rods that produces the sculpture with as small a total imbalance as you can. Note that **you are not required to find the best possible solution**.

Input

You are given twelve input files `00.in`, `01.in`, ..., `11.in`, each describing a collection of wooden blocks. You can download these files as a Zip Archive from the Statement page for this problem. Please make sure you can do so and open and extract the input files (right click) from the zip, as you will be **unable to attempt this problem** without opening these files! *Contact the judges via the Communication page if you have any difficulties with this.*

In each input file:

- The first line contains a single integer N, the number of blocks.
- The following N lines contain one integer each. The i-th of which is w_i, the weight of the i-th block.

Output

For each input file `xy.in`, you must produce an output file `output_xy.txt` that contains your solution.

Each solution will be given by a string on a single line, describing the arrangement of your sculpture. This line must consist only of digits, `(`, and `)`. In particular, a block of weight `w` is represented by `(w)`, and a hook with two sub-sculptures `a` and `b` hanging from it is represented by `(ab)`.

Thus, a single rod with a weight 1 and a weight 2 block hanging from its hooks would be denoted by `((1)(2))`. If this in turn was hung from a hook on a rod with a weight 3 block hanging from the other hook, the sculpture would be given by `(((1)(2))(3))`.

Sample Input

```
4
10
20
30
40
```

Possible Output 1

```
(((40)(10))((20)(30)))
```

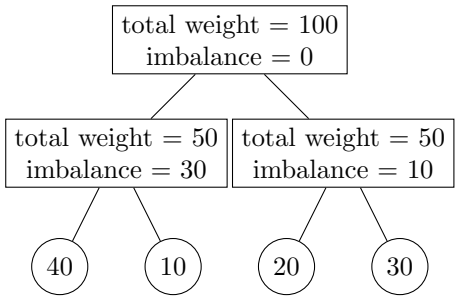
Possible Output 2

```
((((20)(10))(30))(40))
```

Explanation of Sample Input

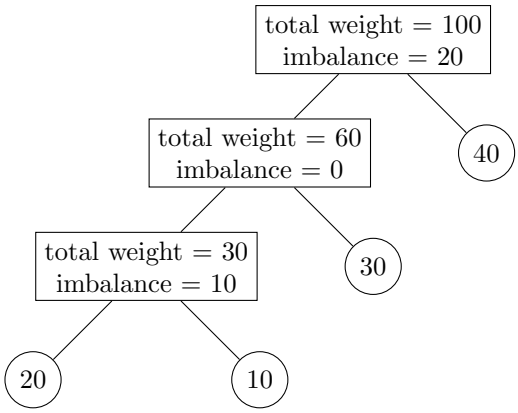
This Sample Input file corresponds to `00.in`.

Explanation of Possible Output 1



The first possible output represents a rod from which two other rods hang. One of these rods holds blocks with weights 10 and 40, while the other holds blocks with weights 20 and 30. The former rod has an imbalance of 30 since $|40 - 10| = 30$, for the same reason the latter rod has an imbalance of $|20 - 30| = 10$. Since both of these rods carry 50 weight in total, the topmost rod has no imbalance at all. The total imbalance of this sculpture is thus $30 + 10 + 0 = 40$.

Explanation of Possible Output 2



For the sculpture described by the second possible output, we find rods with imbalances 10, 0, and 20, and it therefore has a total imbalance of $10 + 0 + 20 = 30$.

Constraints

For all cases, $1 \leq N \leq 10\,000$ and $1 \leq w_i \leq 100\,000$.

Submission

In any submission, you may submit output files to some or all of the input files. You may do so by either uploading each output file on the Submission page for this problem, or by creating a zip archive containing your output files, and uploading this to the Submission page.

On a Unix system you can use a command like the following to create a zip:

```
zip mysolutions.zip output_*.txt
```

On Windows systems you can create a zip by selecting *File → New → Compressed (zipped) Folder* from within Windows Explorer, and then you can copy your output files into this new zip file.

If a submission lacks the output for a certain testcase, the current submission is completed with the **most recently submitted output for that testcase** (if it exists). This means you can work on a single testcase at a time and only need to submit for testcases you have worked on since your last submission.

Contact the judges via the Communication page if you have any difficulties zipping or making a submission.

Scoring

You are highly encouraged to look at the **details** *window for each of your submissions on the contest website.*

For each output file, your score will be computed as follows. If the output file does not describe a valid sculpture, you will score 0. In this case, the contest system will display either `Invalid output format` or `Invalid weights` (if an incorrect or incomplete set of weights are present) in the Details column for this testcase. Otherwise, the details column will display `Valid output` and your score is calculated according to the following formula, based on the imbalance of your solution and that of the judges' solution for the same testcase:

$$\text{Your score} = \max\left(0, \min\left(100, 105 - 10\left(\frac{\text{Your imbalance}}{\text{Judges' imbalance}}\right)^2\right)\right)\%$$

In particular, your score is always between 0 and 100 (inclusive) and the judges' solution scores 95% on each testcase. Note that if your imbalance is rather large, you may still score 0 with a `Not correct` outcome, so **please check details** to ensure that your output is valid.

Each subtask contains a single testcase, so your score for each subtask is your score on that testcase multiplied by the number of points for that subtask. Points are distributed across subtasks as follows.

Subtask	Input file	Output file	N	Judges' imbalance	Points
1	`00.in`	`output_00.txt`	4	30	1
2	`01.in`	`output_01.txt`	10	117	4
3	`02.in`	`output_02.txt`	100	961791	10
4	`03.in`	`output_03.txt`	1000	10848932	10
5	`04.in`	`output_04.txt`	10000	129746831	10
6	`05.in`	`output_05.txt`	16	64	5
7	`06.in`	`output_06.txt`	128	33216	10
8	`07.in`	`output_07.txt`	1024	260952	10
9	`08.in`	`output_08.txt`	8192	2049420	10
10	`09.in`	`output_09.txt`	100	82788	10
11	`10.in`	`output_10.txt`	1000	652093	10
12	`11.in`	`output_11.txt`	10000	6392613	10

Your score for each submission will be the sum of your scores for each subtask, rounded to 2 decimal places. Note that your score for this problem is the **maximum among all your submissions**, so you may wish to ensure that the best scoring output files for each input file are submitted together in a single submission before the end of the contest.