Task: SLI
Sliding Cat Puzzle

AACPP SuSe 2025 Round 1 Memory: 64MiB 2025.05.13 – 2025.05.27

Dexter the Cat has a new toy, called the Sliding Cat Puzzle. It is built as a row of n slots. Dexter's human puts one treat into each slot and then covers them all with a special mechanism. The mechanism exposes a single hole through which Dexter can access and extract the treat. There is also a lever that Dexter can push left or right, causing the hole to move to a slot to the left or the right of the current one. The objective is to, of course, get all the treats.

Dexter loves treats, but he's also very lazy. It takes quite a bit of effort to get the lever moving at all. This effort depends on whether Dexter is pushing left or right and it changes during the game. Once it gets moving it proceeds smoothly, so there is no difference in effort between moving the lever one slot or multiple slots in a single move. The total effort required from Dexter is the sum of these costs over all $n-1$ required moves in the game.

Knowing the effort costs for each move and that the game starts with the hole over slot s help Dexter determine the **minimal effort** required to get all treats.

Input

In the first line of standard input there are two integers n, s, the number of slots and the index of the initially open slot counting from the left ($1 \leq s \leq n$). Then, $n-1$ lines follow. Each contains two integers, l_i, r_i, describing the effort required to push the lever as the i-th move, to the left or right, respectively.

Output

Your program should write two lines to standard output. The first line should contain a single integer – the minimal effort required to get all the treats. The second line should contain the strategy that achieves this minimum – n integers s_i describing over which slot the hole is located in the i-th step of the game. There might be more than one such optimal strategy, in which case your program can output any one of them.

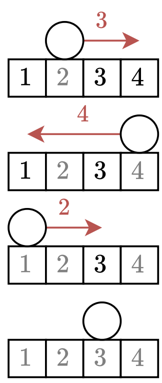
Example

For the input:

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

the correct output is:

```
9
2 4 1 3
```



The game starts with the hole over slot 2. Dexter then pushes the lever two slots to the right (over slot 4), which costs 3 effort. He then pays 4 effort to move three slots to the left over slot 1. Finally, he pushes the lever right by two slots paying 2 effort, which finishes the game. In total the cost is $3+4+2=9$. There are other strategies that achieve effort 9 or lower, e.g. $2, 3, 1, 4$.

Additional examples

The following initial tests are also available:

- `0b` – $n = 10$, $s = 1$, $l_i = 1$, $r_i = 2$ for all i;
- `0c` – $n = 18$, $s = 7$, $l_i = i, r_i = i + 1$ for odd i, $l_i = i + 1, r_i = i$ for even i;
- `0d` – $n = 500$, $s = 250$, $l_i = 0$, $r_i = 1$ for odd i, $l_i = 1, r_i = 0$ for even i;
- `0e` – $n = 3\,000$, $s = 1\,000$, $l_i = r_i = i$ for all i;
- `0f` – $n = 500\,000$, $s = 1$, $l_i = i$, $r_i = 500\,000 - i$ for all i.

Limits

Your solution will be evaluated on a number of hidden test cases divided into groups. Points for a group are awarded if and only if the submission returns the correct answer for each of the tests in the group within the allotted time limit. These groups are organised into subtasks with the following limits and points awarded.

In all tests $1 \leq s \leq n$ and $0 \leq l_i, r_i \leq 10^6$.

Partial points

If your solution outputs the optimal cost (first line of output), and the second line is left blank or not correct, it will receive 50% of the points for a given test group.

Subtask	Limits	Points
1.	$1 \leq n \leq 500$	2
2.	$1 \leq n \leq 3\,000$	2
3.	$1 \leq n \leq 500\,000$, $s = 1$	2
4.	$1 \leq n \leq 500\,000$	4