Task: GWI
Stars

XXVI OI, Stage II, Day one. Source file `gwi.*` Available memory: 256 MB. *13.02.2019*

In a distant perfectly geometric galaxy there are n colinear stars, numbered from 1 to n "left-to-right" in a fixed orientation of the underlying line. Phenomenal devices known as teleporters, widespread among the galaxy's inhabitants, allow instantaneous travel between the stars.

Bytalene, who lives on star no. s, has recently bought her first teleporter. To celebrate, she plans to perform a sequence of $n-1$ teleportations, in order to visit every star in the galaxy exactly once (star s counts as visited right from the start). She would like to do so in a way that minimizes energy expenditure, as charging a teleporter's battery is extremely expensive.

The energy required for teleportation obeys rules that are well-known to physicists but little understood by mere mortals such as Bytalene: the energy is determined by the direction one travels (leftward, to a star with lower number, or rightward, to one with a higher number) and the number of teleportations made so far with the device. Exact formulas are fairly complex, but helpfully the teleporter's manual contains the values for the i-th teleportation, l_i if it is leftward and r_i if it is rightward, for $1 \le i \le n-1$. (In fact, the manual lists them for larger values of i as well, but Bytalene is not interested in those.)

Your task is to help Bytalene find the cheapest way of visiting all n stars.

Input

In the first line of the standard input, there are two integers n and s ($n \ge 2$, $1 \le s \le n$), separated by a single space, which specify the number of stars in the galaxy and the number of the star where Bytalene lives.

The $n-1$ lines that follow list the teleportation costs; the i-th such line contains two integers l_i, r_i ($0 \le l_i$, $r_i \le 10^6$), separated by a single space, which specify the cost of the i-th teleportation depending on its direction.

Output

A single integer should be printed to the standard output: the minimum total cost of the whole sequence of teleportations. In the second line, a sequence of n pairwise different integers from the range $[1, n]$ should be printed, separated by single spaces. These should be the numbers of the stars to be visited in sequence, starting with s. Your program can report any optimal solution, should there be more than one.

Example

For the input data:

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

the correct result is:

```
9
2 4 1 3
```

Explanation for the example: Bytalene begins her journey on star no. 2. Her first teleportation is rightward (to the star no. 4), which costs $r_1 = 3$. The second teleportation is leftward (to the star no. 1), which costs $l_2 = 4$. Finally, the last teleportation, which brings Bytalene to the star no. 3, costs $r_3 = 2$. (Notice that $l_3 = r_3$, so the last teleportation would cost 2 regardless of direction.)

The total cost is $3 + 4 + 2 = 9$, which is optimal.

Sample grading tests:

- `1ocen`: $n = 10$, $s = 1$; $l_i = 1$, $r_i = 2$ for every i;
- `2ocen`: $n = 18$, $s = 7$; $l_i = i$, $r_i = i + 1$ for odd i; $l_i = i + 1$, $r_i = i$ for even i;
- `3ocen`: $n = 500$, $s = 250$; $l_i = 0$, $r_i = 1$ for odd i; $l_i = 1$, $r_i = 0$ for even i;
- `4ocen`: $n = 3000$, $s = 1000$; $l_i = r_i = i$ for every i;
- `5ocen`: $n = 500\,000$, $s = 1$; $l_i = i$, $r_i = 500\,000 - i$ for every i.

Grading

The set of tests consists of the following subsets. Within each subset, there may be several unit tests.

Time limits for each subset are published in SIO.

Subset	**Condition**	**Score**
1	$n \leq 10$	8
2	$n \leq 18$	8
3	$n \leq 500$	10
4	$n \leq 3000$	16
5	$n \leq 500\,000$; $l_i \leq r_i$ for every i	10
6	$n \leq 500\,000$; optimal cost is 0 and for all i exactly one of l_i, r_i is 0	10
7	$n \leq 500\,000$; $s = 1$	18
8	$n \leq 500\,000$	20

In subsets 1, 2, 3, 4, 5, 7, and 8, you will receive 50% of the points for each test where you report correct optimal cost but incorrect sequence.