



Sorting

Aizhan has a sequence of N integers $S[0], S[1], \dots, S[N - 1]$. The sequence consists of distinct numbers from 0 to $N - 1$. She is trying to sort this sequence in ascending order by swapping some pairs of elements. Her friend Ermek is also going to swap some pairs of elements — not necessarily in a helpful way.

Ermek and Aizhan are going to modify the sequence in a series of rounds. In each round, first Ermek makes a swap and then Aizhan makes another swap. More precisely, the person making a swap chooses two valid indices and swaps the elements at those indices. Note that the two indices do not have to be distinct. If they are equal, the current person swaps an element with itself, which does not change the sequence.

Aizhan knows that Ermek does not actually care about sorting the sequence S . She also knows the exact indices Ermek is going to choose. Ermek plans to take part in M rounds of swapping. We number these rounds from 0 to $M - 1$. For each i between 0 and $M - 1$ inclusive, Ermek will choose the indices $X[i]$ and $Y[i]$ in round i .

Aizhan wants to sort the sequence S . Before each round, if Aizhan sees that the sequence is already sorted in ascending order, she will terminate the entire process. Given the original sequence S and the indices Ermek is going to choose, your task is to find a sequence of swaps, which Aizhan can use to sort the sequence S . In addition, in some subtasks you are required to find a sequence of swaps that is as short as possible. You may assume that it is possible to sort the sequence S in M or fewer rounds.

Note that if Aizhan sees that the sequence S is sorted after Ermek's swap, she can choose to swap two equal indices (e.g., 0 and 0). As a result the sequence S is also sorted after the entire round, so Aizhan reaches her goal. Also note that if the initial sequence S is already sorted, the minimal number of rounds needed to sort it is 0 .

Example 1

Suppose that:

- The initial sequence is $S = 4, 3, 2, 1, 0$.
- Ermek is willing to make $M = 6$ swaps.
- The sequences X and Y that describe the indices Ermek is going to choose are $X = 0, 1, 2, 3, 0, 1$ and $Y = 1, 2, 3, 4, 1, 2$. In other words, the pairs of indices that Ermek plans to choose are $(0, 1)$, $(1, 2)$, $(2, 3)$, $(3, 4)$, $(0, 1)$, and $(1, 2)$.

In this setting Aizhan can sort the sequence S into the order $0, 1, 2, 3, 4$ in three rounds. She can do so by choosing the indices $(0, 4)$, $(1, 3)$, and then $(3, 4)$.

The following table shows how Ermek and Aizhan modify the sequence.

Round	Player	Pair of swapped indices	Sequence
beginning			4, 3, 2, 1, 0
0	Ermek	(0, 1)	3, 4, 2, 1, 0
0	Aizhan	(0, 4)	0, 4, 2, 1, 3
1	Ermek	(1, 2)	0, 2, 4, 1, 3
1	Aizhan	(1, 3)	0, 1, 4, 2, 3
2	Ermek	(2, 3)	0, 1, 2, 4, 3
2	Aizhan	(3, 4)	0, 1, 2, 3, 4

Example 2

Suppose that:

- The initial sequence is $S = 3, 0, 4, 2, 1$.
- Ermek is willing to make $M = 5$ swaps.
- The pairs of indices that Ermek plans to choose are $(1, 1)$, $(4, 0)$, $(2, 3)$, $(1, 4)$, and $(0, 4)$.

In this setting Aizhan can sort the sequence S in three rounds, for example by choosing the pairs of indices $(1, 4)$, $(4, 2)$, and then $(2, 2)$. The following table shows how Ermek and Aizhan modify the sequence.

Round	Player	Pair of swapped indices	Sequence
beginning			3, 0, 4, 2, 1
0	Ermek	(1, 1)	3, 0, 4, 2, 1
0	Aizhan	(1, 4)	3, 1, 4, 2, 0
1	Ermek	(4, 0)	0, 1, 4, 2, 3
1	Aizhan	(4, 2)	0, 1, 3, 2, 4
2	Ermek	(2, 3)	0, 1, 2, 3, 4
2	Aizhan	(2, 2)	0, 1, 2, 3, 4

Task

You are given the sequence S , the number M , and the sequences of indices X and Y . Compute a sequence of swaps, which Aizhan can use to sort the sequence S . In subtasks 5 and 6 the sequence of swaps you find has to be the shortest possible.

You need to implement the function `findSwapPairs`:

- `findSwapPairs(N, S, M, X, Y, P, Q)` — This function will be called by the grader exactly once.
 - N : the length of the sequence S .
 - S : an array of integers containing the initial sequence S .

- M : the number of swaps Ermek plans to make.
- X, Y : arrays of integers of length M . For $0 \leq i \leq M - 1$, in round i Ermek plans to swap numbers at indices $X[i]$ and $Y[i]$.
- P, Q : arrays of integers. Use these arrays to report one possible sequence of swaps Aizhan can make to sort the sequence S . Denote by R the length of the sequence of swaps that your program has found. For each i between 0 and $R - 1$ inclusive, the indices Aizhan should choose in round i should be stored into $P[i]$ and $Q[i]$. You may assume that the arrays P and Q have already been allocated to M elements each.
- This function should return the value of R (defined above).

Subtasks

subtask	points	N	M	extra constraints on X, Y	requirement on R
1	8	$1 \leq N \leq 5$	$M = N^2$	$X[i] = Y[i] = 0$ for all i	$R \leq M$
2	12	$1 \leq N \leq 100$	$M = 30N$	$X[i] = Y[i] = 0$ for all i	$R \leq M$
3	16	$1 \leq N \leq 100$	$M = 30N$	$X[i] = 0, Y[i] = 1$ for all i	$R \leq M$
4	18	$1 \leq N \leq 500$	$M = 30N$	none	$R \leq M$
5	20	$6 \leq N \leq 2,000$	$M = 3N$	none	minimum possible
6	26	$6 \leq N \leq 200,000$	$M = 3N$	none	minimum possible

You may assume that there exists a solution that requires M or fewer rounds.

Sample grader

The sample grader reads the input from the file `sorting.in` in the following format:

- line 1: N
- line 2: $S[0] \dots S[N - 1]$
- line 3: M
- lines 4, ..., $M + 3$: $X[i] \ Y[i]$

The sample grader prints the following output:

- line 1: the return value R of `findSwapPairs`
- line $2+i$, for $0 \leq i < R$: $P[i] \ Q[i]$