Tennis tournament

Each year Enfield Tennis Academy (ETA) juniors play a tournament against their rivals from Port Washington Academy.

As last year's winner, Port Washington is the host of this year's competition – in compensation for this, ETA head coach Gerhardt Schtitt is allowed to decide on the match schedule, i.e. who plays whom. More precisely, each of the academies sends a team of N players ($1 \le N \le 750\,000$) and there will be N matches, each between one ETA player and one player from Port Washington, in such a way that everyone plays exactly one match. The team that wins the majority of matches will be declared the overall winner.

Since this title is quite prestigious (and on the other hand the loser is traditionally shamed), Schtitt wants to choose the match pairings cleverly. Fortunately he has the current junior rankings at hand and he is sure that these determine the outcome of each match: a player wins against another player if and only if he is ranked higher (i.e. with a *lower* ranking number) than the other. In particular, the current No. 1 would win against any other player. We further assume that there are no ties in the ranking.

Help Schtitt by writing a program that, given the player rankings for each team, decides which matches Schtitt should announce such that his team wins as much matches as possible.

Input

The first line of input contains the integer N with the meaning described above. The next line consists of N space separated integers r_i ($1 \le r_i \le 10^9$): the rankings of the ETA players. The final line describes the rankings of the Port Washington players in the same way.

Output

The first line of output should contain the maximum number of matches that the ETA team can win. After that you should output N lines, describing an optimal match schedule. Here each of these lines should consist of two space separated integers a and b ($1 \le a, b \le n$), meaning that the a^{th} ETA player should play against the b^{th} player from Port Washington (according to the ordering in the input file, using 1-based indexing).

If there are multiple solutions, you may output any one of them.

Constraints and grading

We always have $1 \le N \le 750\,000$ and $1 \le r_i \le 10^9$. Moreover, no ranking occurs twice in the input (there are no ties).

Subtask 1 (20 points). $N \le 10$.

Subtask 2 (40 points). $N \le 10\,000$.

Subtask 3 (40 points). No further constraints.

Moreover we have: if for all testcases of a given subtask the first line of your output is correct (but at least in one of these testcases the rest of the output isn't) you'll be awarded 75% of points for this subtask.

Samples

Input	Output

3
6 3 4
2 8 5
	2
	1 1
	2 2
	3 3

3
4 5 6
1 2 3
	0
	1 1
	2 2
	3 3

Limits

Time: 0.7 s
Memory: 128 MiB

Feedback

restricted feedback is given for this task, i.e. the score shown equals your real score on this submission. However, for each testcase group you are only shown the verdict for the first testcase with minimal score in that group. (The order of the testcases in each of the groups is fixed.)

Scoring

Your final score for this task is the sum over each subtask of your best score on that subtask across all your submissions.