Engineering

Winter Workshops, Day 5. Memory limit: 512 MB. *08.01.2020*

Paweł has been promoted and he is now leading a team of n software engineers in a complex web app project. Having divided the project into k simpler tasks he now has to assign them to his subordinates. The engineers were busy watching *Block Crew*, writing 100 lines of commit messages or debugging the code force-pushed to master branch instead of actually working, so a lot still needs to be done and the deadline is approaching quickly. No employee can be assigned to more than one task, but each task still needs to be assigned to exactly one employee, because the developers are obviously unable to cooperate with each other.

Paweł is a good boss, so he asked each of the engineers which task they would like to take, the i-th one selected task a_i. Unfortunately, some tasks might have been selected more than once (frontend), while others (machine learning), might have not been selected at all! Our team leader can also ask i-th subordinate to work on a task other than a_i, but if that is the case, in order to keep them happy, he needs to pay them b_i\$ in benefits. Help Paweł to cheaply finish the project and print the minimum amount of \$ he needs to pay in benefits to complete all the tasks.

Constraints

- $1 \leq k \leq n \leq 10^5$
- $1 \leq a_i \leq k$
- $1 \leq b_i \leq 10^9$

Input

$n\ k$ $a_1\ a_2\ ...\ a_n$ $b_1\ b_2\ ...\ b_n$

Output

The first line of output should contain the minimal amount of \$ Paweł has to pay, in order to complete the project.

Scoring

Subtask	**Constraints**	**Points**
1	$n < 20$	10
2	$n < 2000$	20
3	$b_i < 10^5$	30
4	no additional constraints	40

Example

Input	Output
8 7 1 1 3 1 5 3 7 1 5 7 4 8 1 3 5 2	10
3 3 3 1 2 5 3 4	0
14 6 2 3 6 6 6 2 5 3 5 5 5 6 5 5 56 41 19 81 19 32 99 37 97 20 90 44 19 79	38