

All Wrapped Up

December 2021

C++ — 1 SEC — 512 MB

M.O.T.H.E.R., the chip-manufacturing behemoth, has decided to reward all its employees this Christmas. Usually, each employee gets a novelty Santa hat and an oversized carrot, but this year they are trying something different.

Each employee is offered n presents and are allowed to select k of them. This year, M.O.T.H.E.R. has produce 3,001 copies of its latest gift-wrapping machine and wants to make the most of them. For this reason, each present has been wrapped an exorbitant number of times. They wish to reuse this wrapping paper, so a present must be fully unwrapped before an employee can take it home. It's the last day before the holiday and everyone wants to get home, so each employee wants to unwrap as few layers of wrapping paper as possible, whilst still obtaining k presents.

INPUT You will be given integers n and k , denoting the number of presents an employee is offered, and how many they may take, respectively. The next line will contain n numbers, representing the number of layers of wrapping paper the i th present is wrapped in. However, the presents have all been jumbled up, so the order is now useless to you.

$$1 \leq n \leq 20,000$$

$$1 \leq k \leq n$$

OUTPUT Output a single integer, the minimum number of wrapping paper layers that need to be unwrapped in order for the employee to claim k presents, in the worst case.

SAMPLES For example, suppose an employee is offered 5 presents and is allowed to select 3 of them. The presents are wrapped in 1, 2, 3, 4, and 5 layers, but it is not known which present is which. If they unwrap 3 layers from each present, they will have to unwrap 12 layers in total in the worst case.

INPUT

5 3
1 2 3 4 5

6 5
10 30 14 26 3 13

4 3
3 3 5 5

OUTPUT

12

84

13

