You are given a **0-indexed** integer array candies. Each element in the array denotes a pile of candies of size candies[i]. You can divide each pile into any number of **sub piles**, but you **cannot** merge two piles together.

You are also given an integer k. You should allocate piles of candies to k children such that each child gets the **same** number of candies. Each child can take **at most one** pile of candies and some piles of candies may go unused.

Return *the****maximum number of candies****each child can get.*

**Example 1:**

**Input:** candies = [5,8,6], k = 3

**Output:** 5

**Explanation:** We can divide candies[1] into 2 piles of size 5 and 3, and candies[2] into 2 piles of size 5 and 1. We now have five piles of candies of sizes 5, 5, 3, 5, and 1. We can allocate the 3 piles of size 5 to 3 children. It can be proven that each child cannot receive more than 5 candies.

**Example 2:**

**Input:** candies = [2,5], k = 11

**Output:** 0

**Explanation:** There are 11 children but only 7 candies in total, so it is impossible to ensure each child receives at least one candy. Thus, each child gets no candy and the answer is 0.

**Constraints:**

* 1 <= candies.length <= 105
* 1 <= candies[i] <= 107
* 1 <= k <= 1012