36m left

⌘

ALL

ⓘ

1

2

3

3. Efficient Study

An e-book provider has a list of programming articles, each with a different *intellectual value*. As an incentive, readers receive awards based on their scores. Readers get points equal to an article's intellectual value for each article they have read twice. Given a list of articles with their page lengths, each article's intellectual value coefficient, and a limited number of pages that can be read in a day, determine the maximum achievable intellectual value from reading articles during one day.

Example

iv = [2, 4, 4, 5]

articles = [2,2,3,4]

p = 15

There are *n = 4* articles *= [2,2,3,4] that have a corresponding* intellectual values *iv = [2, 4, 4, 5]*.

- *p = 15* pages max can be read per day.
- Two best approaches to read the maximum articles are: Read the first, second and third articles (*2 * articles[0] + 2 * articles[1] + 2 * articles[2] = 2 * 2 + 2 * 2 + 2 *3 = 14* pages) or read the third and fourth articles (*2 * articles[2] + 2 * articles[3] = 2 * 3 + 2 * 4 = 14* pages) .
- The maximum total intellectual value is: *iv[0] + iv[1] + iv[2] = 2 + 4 + 4 = 10*.

Function Description

Complete the function *maximumLearning* in the editor below.

maximumLearning has the following parameter(s):

int iv[n]: an array of integers, each article's intellectual value

int articles[n]: an array of integers, each article's page length

int p: number of pages that can be read in one day

Returns:

int: integer value representing the maximum achievable intellectual value in one day of reading

Constraints

- $1 \le n \le 10^3$
- $1 \le iv[i] \le 10^6$, where $0 \le i < n$.
- $1 \le articles[i] \le 100$, where $0 \le i < n$.
- $1 \le p \le 10^3$

Input Format for Custom Testing