

40m left

2. Code Question 2



ALL



2

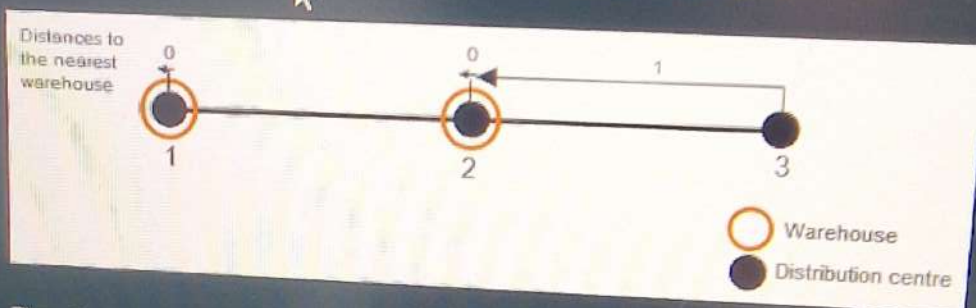
Amazon has recently established n distribution centers in a new location. They want to set up 2 warehouses to serve these distribution centers. Note that the centers and warehouses are all built along a straight line. A distribution center has its demands met by the warehouse that is closest to it. A logistics team wants to choose the location of the warehouses such that the sum of the distances of the distribution centers to their closest warehouses is minimized.

Given an array `dist_centers`, that represent the positions of the distribution centers, return the minimum sum of distances to their closest warehouses if the warehouses are positioned optimally.

Example

Suppose `dist_centers = [1, 2, 3]`.

One optimal solution is to position the 2 warehouses at $x_1 = 1$ and $x_2 = 2$.



The minimum sum of the distances between distribution centers and the warehouses closest to them is $0 + 0 + 1 = 1$.

Function Description

Complete the function `getMinTotalDistance` in the editor below.

`getMinTotalDistance` has the following parameter:

`int dist_centers[n]`: the locations of the distribution centers

Returns

`int`: the minimum sum of the given expression

Constraints

- $1 \leq n \leq 2 \cdot 10^3$
- $0 \leq \text{dist_centers}[i] \leq 10^6$

► Input Format For Custom Testing

▼ Sample Case 0

Sample Input For Custom Testing

STDIN FUNCTION

2 1 2 3



21°C
Partly sunny

Test R

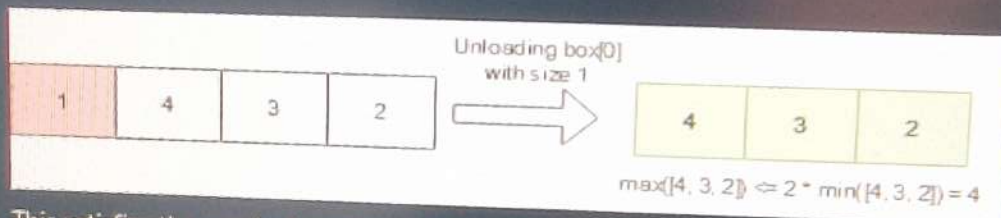
1. Code Question 1

The supply chain manager at one of Amazon's warehouses is shipping the last container of the day. All n boxes have been loaded into the truck with their sizes represented in the array *boxes*. The truck may not have enough capacity to store all the boxes though, so some of the boxes may have to be unloaded. The remaining boxes must satisfy the condition $\max(\text{boxes}) \leq \text{capacity} * \min(\text{boxes})$.

Given the array, *boxes*, and *capacity*, find the minimum number of boxes that need to be unloaded.

Example

Given, $n = 4$, boxes having size *boxes* = [1, 4, 3, 2] and *capacity* = 2.



This satisfies the required condition. Hence the answer is 1.

Function Description

Complete the function *getMinimumBoxes* in the editor below.

getMinimumBoxes has the following parameter(s):

int boxes[n]: the size of each box

int capacity: the multiplier

Returns

int: the minimum boxes to remove from the truck

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq \text{boxes}[i] \leq 5 \times 10^5$
- $1 \leq \text{capacity} \leq 1000$

► Input Format For Custom Testing

▼ Sample Case 0

Sample Input For Custom Testing

STDIN	FUNCTION
6	boxes[] size n = 6
4	boxes[] = [4, 5, 3, 8, 3, 7]
5	
3	
8	
3	
7	
2	