**Write computer programs for the following algorithms/problems:**

1. **Given a sorted array of integers a, find an integer x from a such that the value of**

abs(a[0] - x) + abs(a[1] - x) + ... + abs(a[a.length - 1] - x)

**is the smallest (here abs denotes the absolute value). If there are several possible answers, output the smallest one.**

Example:

For a = [2,4,7], the output should be absoluteValuesSumMinimization(a) = 4.

For a = [2,4,7,6], the output should be absoluteValuesSumMinimization(a) = 4.

Notes:

[input] integer array a (non-empty array of integers)

[output] integer

1<= a.length <=200

Example tests: [2,4,7,6,6] , [2,4,7,6,6,8]

1. **Given a integer n, return the sum of its digits.**

Example:

For n=29, the output should be addDigits(n) = 11.

1. **Given an array of integers, find the pair of adjacent elements that has the largest product and return the product.**

Example:

For inputArray = [3, 6, -2, -5, 7, 3], the output should be adjacentElementsProduct(inputArray) = 21.

7 and 3 produce the largest product.

1. **Several people are standing in a row and need to be divided into two teams. The first person goes into team 1, the second goes into team 2, the third goes into team 1 again, the fourth goes into team 2, and so on. You are given an array of positive integers (the weights of the people). Return an array of two integers, where the first element is the total weight of team 1, and the second element is the total weight of team 2 after the division is complete.**

Example:

For a = [50, 60, 60, 45, 70], the output should be alternatingSums(a) = [180, 105].

1. **Given an array of 2k integers (for some integer k), perform the following operations until the array contains only one element:**

* **On the 1st, 3rd, 5th, etc. iterations, replace each pair of consecutive elements with their sum.**
* **On the 2nd, 4th, 6th, etc. iterations, replace each pair of consecutive elements with their product.**

**After the algorithm has finished, there will be a single element left in the array. Return that element.**

Example:

For inputArray = [1, 2, 3, 4, 5, 6, 7, 8], the output should be arrayConversion(inputArray) = 186.

Notes: [1, 2, 3, 4, 5, 6, 7, 8] => [3, 7, 11, 15] => [21, 165] => [186] => 186

1. **Given array of integers, for each position I, search among the previous position (from the left) that contains a smaller value. Store this value at position I in the answer. If no such value can be found, store -1 instead.**

For inputArray = [3, 5, 2, 4, 5], the output should be arrayPreviousLess(inputArray) = [-1 ,3, -1, 2, 4].

1. **Given a rectangular matrix of characters, add a border of asterisks(\*) to it.**

Example:

For picture=[“abc”,”ded”]

The output should be

addBorder(picture) = [“\*\*\*\*\*”,

“\*abc\*”,

“\*ded\*”,

“\*\*\*\*\*” ]

Notes:

[input] string array picture

1. **You are given an array of integers representing coordinates of obstacles situated on a straight line.**

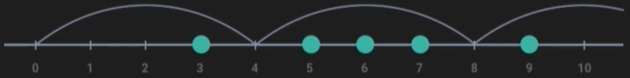
**Assume that you are jumping from the point with coordinate 0 to the right. You are allowed only to make jumps of the same length represented by some integer.**

**Find the minimal length of the jump enough to avoid all the obstacles.**

Example:

For inputArray = [5, 3, 6, 7, 9], the output should be avoidObstacles(inputArray) = 4.

Check out the image below for better understanding:



1. **Given the positions of a white bishop and a black pawn on the standard chess board, determine whether the bishop can capture the pawn in one move.**

**The bishop has no restrictions in distance for each move but is limited to diagonal movement.**

**Check out the example below to see how it can move:**

|  |  |
| --- | --- |
| dama taşı, kozmetik içeren bir resim  Açıklama otomatik olarak oluşturuldu | metin, dama taşı içeren bir resim  Açıklama otomatik olarak oluşturuldu |

Example:

For bishop = “a1” and pawn = “c3”, the output should be bishopAndPawn(bishop, pawn) = true.