Technical assignment

Notes:

- Choose Any programming language suitable for you (Preferably: Java, Js, Python)
- Respect the Data Structure & Function description.
- Hand back the full solution in a Github repository (each test in a one separate file)

1. Assignment 01:

You're given the pointer to the head nodes of two linked lists. Compare the data in the nodes of the linked lists to check if they are equal. If all data attributes are equal and the lists are the same length, return 1. Otherwise, return 0.

- Example:

$$llist1 = 1 \rightarrow 2 \rightarrow 3 \rightarrow NULL$$
$$llist2 = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow NULL$$

The two lists have equal data attributes for the first 3 nodes. llist2 is longer, though, so the lists are not equal. Return 0.

- Data Structure :

```
 /*
* For your reference:
*
* SinglyLinkedListNode {
*     int data;
*     SinglyLinkedListNode next;
* }
*/
```

Function Description

Create the compare_lists function that has the following parameters:

- SinglyLinkedListNode llist1: a reference to the head of a list
- SinglyLinkedListNode llist2: a reference to the head of a list

Returns: - int: return 1 if the lists are equal, or 0 otherwise

Input Format

The first line contains an integer ***t***, the number of test cases.

Each of the test cases has the following format:

- The first line contains an integer ***n***, the number of nodes in the first linked list.
- Each of the next ***n*** lines contains an integer, each a value for a data attribute.

The next line contains an integer ***m***, the number of nodes in the second linked list.

- Each of the next ***m*** lines contains an integer, each a value for a data attribute.

Sample Input:

```
2
2
1
2
1
1
2
1
2
2
1
2
```

Output Format

Compare the two linked lists and return 1 if the lists are equal. Otherwise, return 0.

The output: For each test case, in a new line, print 1 if the two lists are equal, else print 0 .

Sample Output:

```
0
1
```

There are ***t*** = 2 test cases, each with a pair of linked lists.

- In the first case, linked lists are: 1 -> 2 -> NULL and 1 -> NULL
- In the second case, linked lists are: 1 -> 2 -> NULL and 1 -> 2 -> NULL

2. Assignment 02:

The distance between two array values is the number of indices between them. Given ***a***, find the minimum distance between any pair of equal elements in the array. If no such value exists, return -1.

- Example:

$$a = [3, 2, 1, 2, 3]$$

There are two matching pairs of values: 3 and 2.

- The indices of the 3's are $i = 0$ and $j = 4$, so their distance is $d[i, j] = 4$.
- The indices of the 2's are $i = 1$ and $j = 3$, so their distance is $d[i, j] = 2$.

The minimum distance is ***2*** .

- Function Description:

Create the minimumDistances function that has the following parameter(s):

- int a[n]: an array of integers

Returns: - int: the minimum distance found or -1 if there are no matching elements

- Input Format:

The first line contains an integer ***n***, the size of array ***a***.

The second line contains ***n*** space-separated integers $a[i]$.

Sample Input:

```
STDIN           Function
-----           --------
6               arr[] size n = 6
7 1 3 4 1 7     arr = [7, 1, 3, 4, 1, 7]
```

- Output Format

Print a single integer denoting the minimum $d[i, j]$ in ***a*** . If no such value exists, print -1.

Sample Output

```
3
```

There are two pairs to consider:

- a[1] and a[4] are both 1 , so $d[1, 4] = 3$.
- a[0] and a[5] are both 7 , so $d[0, 5] = 5$.

The answer is $min(3, 5) = 3$.