Obstacles:

1: For the unionWithNoDuplicate function, I first set all the duplicated entry to empty string and add all the entries which are not empty string, but I realize that if there are empty strings in the original array, then they will not be added to the result.So I used boolean value instead.

2: For the flipAround function, I first used two variables to go through every entry of the array, but I found it harder to deal with the middle one, so I used one variable ‘n’/2 instead.

Test Case:

testArray1[]={"alpha", "beta", "gamma", "gamma", "beta", "delta"};

testArray2[]={"samwell", "jon", "margaery", "daenerys",  
 "tyrion", "sansa", "howard", "cersei"”jon”};

testArray3[]={"samwell", "jon", "margaery", "daenerys", "tyrion" };

locateMinimum:

(testArray1,0): test when n=0, if it returns -1.

(testArray1,1): test when we only count 1 entry, if it returns 0

(testArray1,6): test if the function works for normal cases

findLastOccurrence:

(testArray2,0,”samwell”): test if n<=0

(testArray2,9,”terry”):test if there is no occurrence in the array

(testArray2,9,”jon”):test if there are multiple occurrences in the array

flipAround:

(testArray3,0):check if don’t flip at all

(testArray3,1):check if only try to flip one entry,

(testArray3,5):check if have odd number of entries

(testArray3,4):check if have even number of entries

hasNoDuplicates:

(testArray2,-1):check if n<0

(testArray2,0):check if n=0

(testArray2,5):check if it works for no-duplicate array

(testArray2,9):check if it works for duplicate array

unionWithNoDuplicates:

(testArray2,9,testArray3,5,output,outputSize):check if it works for two arrays

with several same entries

(testArray2,9,testArray1,5,output,outputSize):check two arrays with no same entries

shiftRight:

(testArray2, 5,6,”temp”):test if amount > n

(testArray3,6,2,”temp):test if it works for normal cases

isInIncreasingOrder:

(testArray3,1): test if it works for normal cases