Example 8: Create a method which accepts an array of integer elements and return the second smallest element in the array

Method Name	getSecondSmallest
Method Description	Get the second smallest element in the array
Argument	int[]
Return Type	int
Logic	Sort the array and return the second smallest element in the array Hint: 1. Convert to ArrayList 2. Use sort method in Collections class

Example 9: Create a method which can perform the following operations on two String objects S1 and S2. The output of each operation should be added to an arraylist and the arraylist should be returned.(Assume S2 is of smaller size)

Examples for below statements are shown in the Logic part

1. Character in each alternate index of S1 should be replaced with S2
2. If S2 appears more than once in S1, replace the last occurrence of S2 in S1 with the reverse of S2, else return S1+S2
3. If S2 appears more than once in S1, delete the first occurrence of S2 in S1, else return S1
4. Divide S2 into two halves and add the first half to the beginning of the S1 and second half to the end of S1.
   Note: If there are odd number of letters in S2, then add (n/2)+1 letters to the beginning and the remaining letters to the end. (n is the number of letters in S2)
5. If S1 contains characters that is in S2 change all such characters to *

Method Name	modifyStrings
Method Description	Perform the above mentioned actions on a String
Argument	String,String
Return Type	Arraylist
Logic	Do the above mentioned actions on the entered String. For Example S1=”JAVAJAVA” S2=”VA’ 1. **VA**A**VA**A**VA**A**VA**A (J replaced with VA, V replaced with VA etc.) 2. JAVAJA**AV** 3. JAJAVA 4. **V**JAVAJAVA**A**

	5. J***J*** **Output:**{" **VA**A**VA**A**VA**A**VA**A"," JAVAJA**AV**"," JAJAVA"," **V**JAVAJAVA**A**","J***J***"}

Example 10: Create a method that accepts a number and modifies it such that the each of the digit in the newly formed number is equal to the difference between two consecutive digits in the original number. The digit in the units place can be left as it is.

Note: Take the absolute value of the difference. Ex: 6-8 = 2

Method Name	modifyNumber
Method Description	Accepts a number and modify it as per the requirement
Argument	int number1
Return Type	int
Logic	Accept a number and modify it such that the each of the digit in the newly formed number is equal to the difference between two consecutive digits in the original number. For example. Input: 45862 Output:13242 **Algorithm**: • Convert number into String • Extract each char using charAt method • Convert char to int and find the difference • Create new StringBuffer object and keep adding the difference • Finally convert StringBuffer to int

Example 18: Create a method which accepts an integer array, reverse the numbers in the array and returns the resulting array in sorted order

Method Name	getSorted
Method Description	Return the resulting array after reversing the numbers and sorting it
Argument	int []
Return Type	int
Logic	Accept and integer array, reverse the numbers in the array, sort it and return the resulting array. Hint 1. Convert the numbers to String to reverse it 2. Use Collection APIs to sort it **Ex:** {12,23,96,45} **Step 1:** Reverse numbers {21,32,69,54} **Step2:** Sort it {21,32,54,69} **Hint**: Use String to reverse number To sort it, Convert array to ArrayList and use Collections.sort

Example 19: Create a method which accepts an integer array and removes all the duplicates in the array. Return the resulting array in descending order

Method Name	modifyArray
Method Description	Remove duplicates
Argument	int []
Return Type	int []
Logic	Remove the duplicate elements in the array and sort it in descending order Hint: 1. Use Collection API (TreeSet) to remove duplicates and sort the result in ascending order 2. Create a new array, iterate through elements in TreeSet and add it in the reverse order