Java Lab Problem Statements

Exercise 1: Create a class with a method which can calculate the sum of first n natural numbers which are divisible by 3 or 5.	Calculate Sum
Method Name	calculateSum
Method Description	Calculates the sum
Argument	int n
Return Type	int sum
Logic	Calculate the sum of first n natural numbers which are divisible by 3 or 5.
Exercise 2: Create a class with a method to find the difference between the sum of the squares and the square of the sum of the first n natural numbers. Method Name	calculateDifference
Method Description	Calculate the difference
Argument	int n
Return Type	int - Sum
Logic	Find the difference between the sum of the squares of the first n natural numbers and the square of their sum. For Example if n is 10,you have to find (1^2+2^2+3^2+….9^2+10^2)- (1+2+3+4+5…+9+10)^2
Exercise 3: Create a class containing a method to create the mirror image of a String. The method should return the two Strings separated with a pipe(|) symbol . Method Name	getImage
Method Description	Generate the mirror image of a String and add it to the existing string.
Argument	String
Return Type	String
Logic	Accepts One String Find the mirror image of the String Add the two Strings together separated by a pipe(|) symbol. For Example Input : EARTH Output : EARTH|HTRAE Hint: Use StringBuffer API (Ex: For this problem reverse method in Stringbuffer can be used) Note: Learn the other APIs in StringBuffer
Exercise 3: Create a class containing a method to create the mirror image of a String. The method should return the two Strings separated with a pipe(|) symbol . Method Name	getImage
Method Description	Generate the mirror image of a String and add it to the existing string.
Argument	String
Return Type	String
Logic	Accepts One String

	Find the mirror image of the String Add the two Strings together separated by a pipe(|) symbol. For Example Input : EARTH Output : EARTH|HTRAE Hint: Use StringBuffer API (Ex: For this problem reverse method in Stringbuffer can be used) Note: Learn the other APIs in StringBuffer
Exercise 4: Create a method to check if a number is an increasing number Method Name	checkNumber
Method Description	Check if a number is an increasing number
Argument	int number
Return Type	boolean
Logic	A number is said to be an increasing number if no digit is exceeded by the digit to its left. For Example : 134468 is an increasing number
Example 5: Create a method to check if a number is a power of two or not Method Name	checkNumber
Method Description	Checks if the entered number is a power of two or not
Argument	int n
Return Type	boolean
Logic	Check if the input is a power of two. Ex: 8 is a power of 2

Example 6: A school offers medals to the students of tenth based on the following criteria
If(Marks>=90) : Gold
If(Marks between 80 and 90) : Silver
If(Marks between 70 and 80) : Bronze
Note: Marks between 80 and 90 means → marks>=80 and marks<90
Write a function which accepts the marks of students as a Hashmap and return the details of the students eligible for the medals along with type of medal.
The input hashmap contains the student registration number as key and mark as value.

The output hashmap should contain the student registration number as key and the medal type as value. Method Name	getStudents
Method Description	Generate the list of students eligible for scholarship
Argument	Hashmap
Return Type	Hashmap
Logic	The method should return the details of the students eligible for the medals along with the medal type.

Example 7: Create a method which accepts a String and replaces all the consonants in the String with the next alphabet.

Note: Consonant refers to all alphabets excluding vowels Method Name	alterString
Method Description	Replace consonants
Argument	String
Return Type	String
Logic	Return the String replacing all the consonants

	with the next character. For Example :JAVA should be changed as KAWA
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. JAVAJAAV 3. JAJAVA 4. VJAVAJAVAA 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

---