Problem8.java

```
package com.jlcindia.strings;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
//P8- IsSubSequence

public class Problem8 {
	static boolean isSubsequence(String main,String sub) {
		int m= main.length();
		int n= sub.length();

		if(m<n)
			return false;

		int i= 0;
		int j=0;

		while(i<m && j<n) {
			if(main.charAt(i)==sub.charAt(j)) {
				j++;
			}
			i++;
		}
		return (j==n);
	}
	public static void main(String[] args) {
	String main = "srinivas";
	String sub = "snvs";
	boolean b= isSubsequence(main,sub);
	System.out.println(b);
	}
}
//Time Complexity - O(n)
// Space Complexity -  O(1)
```

Problem9.java

```
package com.jlcindia.strings;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
//P9- Remove Duplicate Characters
public class Problem9 {
	static String removeDuplicates(String str) {

		int n= str.length();
		int count[] = new int[26];

		for(int i=0;i<n;i++) {
			count[str.charAt(i)-97]++;
		}

		StringBuffer sb= new StringBuffer();
		for(int i=0;i<count.length;i++) {
			if(count[i]!=0) {
				char ch= (char)(i+97);
				sb.append(ch);
			}
		}
			return sb.toString();
	}

	public static void main(String[] args) {
		String str = "bacacdbac";
		String mystr= removeDuplicates(str);
		System.out.println(mystr);
	}
}

//Time Complexity => O(n)
// Space Complexity => O(1)
```

Problem10.java

```
package com.jlcindia.strings;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

//P10- Reverse the Vowels of String

public class Problem10 {

	public boolean isVowel(char ch) {
		boolean b = ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U';
		return b;
	}

	public String reverseVowels(String str) {
		char chArr[] = str.toCharArray();

		int n= chArr.length;
		int start = 0;
		int end = n-1;

		while(start<end) {

			while(start<n && !isVowel(chArr[start])) {
				start++;
			}

			while(end>=0 && !isVowel(chArr[end])) {
				end--;
			}
```

```
                if(start<end) {
                    char temp = chArr[start];
                    chArr[start] = chArr[end];
                    chArr[end] = temp;
                    }

                    start++;
                    end--;

            }

            String resStr= new String(chArr);
            return resStr;
        }

        public static void main(String[] args) {

            String str = "srinivaas";
            String mystr = reverseVowels(str);
            System.out.println(mystr);

        }
}

//Time Complexity - O(n)
// Space Complexity -  O(n)
```