==Assignment Questions:===================
==================

1. Write a Java program to find the longest substring from a given string that

   doesn’t contain any duplicate characters.

=====================Java Code ===============================

```
import java.util.LinkedHashMap;
import java.util.Scanner;

public class longestSubString
{
    static void longestSubstring(String inputString)
    {
        char[] charArray = inputString.toCharArray();

        String longestSubstring = null;

        int longestSubstringLength = 0;

        LinkedHashMap<Character, Integer> charPosMap = new LinkedHashMap<Character, Integer>();

        for (int i = 0; i < charArray.length; i++)
        {
            char ch = charArray[i];

            //If ch is not present in charPosMap, adding ch into charPosMap along with its position....
            if(!charPosMap.containsKey(ch))
            {
                charPosMap.put(ch, i);
            }
            else
            {
                i = charPosMap.get(ch);

                charPosMap.clear();
            }

            //Update the longestSubString...

            if(charPosMap.size() > longestSubstringLength)
            {
                longestSubstringLength = charPosMap.size();

                longestSubstring = charPosMap.keySet().toString();
            }
        }

        System.out.println("Input String :: "+inputString);

        System.out.print("The longest substring :: \t");
        for(int i = 0; i<longestSubstringLength;i++){

            System.out.print(longestSubstring.charAt(i));

        }


    }

    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the longest String :: \t");

        String longeString = sc.nextLine();

        longestSubString.longestSubstring(longeString);

        sc.close();
    }
}
```

2. Write a Program for the Fibonacci series.

0,1,1,2,3,5,8,13,21

```
======================== Java Code =====================================
import java.util.Scanner;

public class fabonakiSeries {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the First Fibonacci Series :: \t");

        int n = sc.nextInt();

        int a=0, b=1;

        System.out.print("0" + " 1" + " ");

        int num =0;

        int count=2;

        while (count<n) {

            num = a+b;

            System.out.print(num+ " ");

            a=b;
            b=num;

            count++;

        }

        sc.close();

    }
}
```

3. Write a program to print all permutations of a string, for example: ‘Cat’.

```
============================== Java Code ==============================
import java.util.Scanner;

public class permutationString{

    //swap function to swap two characters from indices idx and idx2
    public static void swap(char[] arr, int idx, int idx2) {
        char temp = arr[idx];
        arr[idx] = arr[idx2];
        arr[idx2] = temp;
    }

    public static void solve(char[] arr, int idx) {
        if (idx == arr.length - 1) { //Base condition of recursion
            System.out.print(String.valueOf(arr) + " ");
        }

        for (int i = idx; i < arr.length; i++) {
            swap(arr, idx, i);
            solve(arr, idx + 1);
            swap(arr, idx, i);
            //Backtracking: reverting all the elements to their original places
        }
    }

    public static void main(String[] args) {
        try (Scanner sc = new Scanner(System.in)) {
            System.out.print("Enter string to generate its permutations: ");

            String str = sc.next(); //String input from the user

            if (str.length() == 0 || str == null) {

                return;

            }
            solve(str.toCharArray(), 0);

        } catch(Exception e){
```

```
  System.out.println(e);
 }

    }

}
```

4. How to check if two strings are Anagrams?

   Welcome -> ceelmow

============================== Java Code =====================

```
import java.util.Arrays;
import java.util.Scanner;

public class anagram {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the string :: \t");

        String name = sc.nextLine();

        System.out.print("Enter the second string :: \t");

        String name1 = sc.nextLine();

        char arr1[] = name.toCharArray();

        char arr2[] = name1.toCharArray();

        Arrays.sort(arr1);

        Arrays.sort(arr2);

        name = new String(arr1);

        name1 = new String(arr2);

        if(name.equals(name1)){

            System.out.println("These two strings are anagrams.");

        }
        else{

            System.out.println("These two string are not anagram.");

        }

        sc.close();
    }
}
```

5. Implement stack using queue without using an array or linked list.

   a. Push

   b. Pop

   c. Top

```
=========================== Java Code ==============================

import java.util.*;

class stackUsingQueue {

    static class Stack {
        // Two inbuilt queues
        static Queue<Integer> q1 = new LinkedList<Integer>();
        static Queue<Integer> q2 = new LinkedList<Integer>();

        static int csize;

        void push(int x)
        {
            q2.add(x);

            while (!q1.isEmpty()) {

                q2.add(q1.peek());

                q1.remove();

            }

            // swap the names of two queues

            Queue<Integer> q = q1;

            q1 = q2;

            q2 = q;

        }

        void pop()
        {

            // if no elements are there in q1

            if (q1.isEmpty())
```

```
                return;

            q1.remove();

        }

        int top()
        {
            if (q1.isEmpty())
                return -1;
            return q1.peek();

        }

        int size() { return q1.size(); }
    }

    public static void main(String[] args)
    {
        Stack s = new Stack();
        s.push(10);
        s.push(20);
        s.push(30);

        System.out.println("current size: " + s.size());
        System.out.println(s.top());
        s.pop();
        System.out.println(s.top());
        s.pop();
        System.out.println(s.top());

        System.out.println("current size: " + s.size());
    }
}
```

6. Reverse String.

============================ Java Code ===========================

```
import java.util.Scanner;

public class reverseString {
    public static void main(String[] args) {

        System.out.println("Enter a String :: \t");

        Scanner sc = new Scanner(System.in);

        String name = sc.nextLine();


        // for(int i=name.length()-1; i>=0; i--){
        //     System.out.print(name.charAt(i));
        // }

        // Using String Concatination method

        // String rev = "";
        // for(int i=name.length()-1;i>=0;i--){
        //     rev = rev + name.charAt(i);
        // }

        // System.out.println(rev);

        // Using char Array method

        // char a[] = name.toCharArray();
        // for(int i=name.length()-1;i>=0;i--){
        //     System.out.print(a[i]);
        // }

        // Using String Buffer Class

        StringBuffer sb = new StringBuffer(name);

        System.out.println(sb.reverse());

 sc.close();

    }
}
```