

Program 1: Array Partition

```
package ishwarchavan.com;

public class ArrayPartition { //class created
    public static void main(String[] args) { //main program started
        int[] nums = {1,2,3,4};
        System.out.println(arrayPairSum( nums)); //function calling and printing
    }

    public static int arrayPairSum(int[] nums) { //function created
        quickSort(nums, 0, nums.length - 1);
        int sum = 0;
        for(int i = 0; i < nums.length; i+=2) { //loop iterating
            sum += nums[i];
        }
        return sum; //return sum
    }

    static void swap(int[] arr, int i, int j) { // swap function created
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp; //assigning value
    }

    static int partition(int[] arr, int low, int high) { //position function created
        int pivot = arr[high];
        int i = (low - 1);

        for (int j = low; j <= high - 1; j++) { //loop iterating
            if (arr[j] < pivot) {
                i++;
                swap(arr, i, j); //function calling
            }
        }
        swap(arr, i + 1, high);
        return (i + 1); //return
    }

    static void quickSort(int[] arr, int low, int high) { //quickSort created
        if (low < high) {
            int pi = partition(arr, low, high);
            quickSort(arr, low, pi - 1); //function calling
            quickSort(arr, pi + 1, high);
        }
    }
}
```

Program 2: Permutation in string

```
package ishwarchavan.com;

public class PermutationInString { //class created
    public static void main(String[] args) {
        String s1 = "ab";
        String s2 = "eidbaooo";
        System.out.println(checkInclusion(s1,s2));
    }

    public static boolean checkInclusion(String s1, String s2) { //main program
        started

        if(s1.length()>s2.length()){ //if s1 itself is large return false
            return false;
        }

        int[]freq=new int[26]; //store freq of s1 and s2
```

```

        for(int i=0;i<s1.length();i++){           //prepare freq length till s1 length,for s1
character add 1 for s2 char subs 1
            freq[s1.charAt(i)-'a']+=1;
            freq[s2.charAt(i)-'a']-=1;
        }

        for(int i=0;i<s2.length()-s1.length();i++){ //loop till exhaust at that time
we'll compare total s2.length()characters with s1.length() characters

            if(checkAnagram(freq)){           //check if s1 & s2 are anagram of each other
                return true;
            }

            freq[s2.charAt(i)-'a']+=1;           // if not updated frequently, map to remove s2
first character and add s2 next character
            freq[s2.charAt(i+s1.length())-'a']-=1;
        }

        return checkAnagram(freq);           //check for last comparison
    }

    private static boolean checkAnagram(int[]freq){           //check if freq map has all
zero, it will be true for anagram
        for(int i=0;i<freq.length;i++){
            if(freq[i]!=0){
                return false;
            }
        }
        return true;
    }
}

```

Program 3: Subtree of another tree

```

package ishwarchavan.com;

class Node {           //A binary tree node
    int data;
    Node left, right, nextRight;

    Node(int item)
    {
        data = item;
        left = right = nextRight = null;
    }
}

public class SubtreeOfAnotherTree {           //class created
    Node root1, root2;

    boolean areIdentical(Node root1, Node root2){

        if (root1 == null && root2 == null)           // base cases
            return true;

        if (root1 == null || root2 == null)
            return false;

        return (root1.data == root2.data && areIdentical(root1.left, root2.left)&&
areIdentical(root1.right, root2.right));
    }

    boolean isSubtree(Node T, Node S){

        if (S == null)           // base cases
            return true;

```

```

        if (T == null)
            return false;

        if (areIdentical(T, S))
            return true;

        return isSubtree(T.left, S) || isSubtree(T.right, S);
    }

    public static void main(String args[]){    //main program started
        SubtreeOfAnotherTree tree = new SubtreeOfAnotherTree();

        tree.root1 = new Node(26);
        tree.root1.right = new Node(3);
        tree.root1.right.right = new Node(3);
        tree.root1.left = new Node(10);
        tree.root1.left.left = new Node(4);
        tree.root1.left.left.right = new Node(30);
        tree.root1.left.right = new Node(6);

        tree.root2 = new Node(10);
        tree.root2.right = new Node(6);
        tree.root2.left = new Node(4);
        tree.root2.left.right = new Node(30);

        if (tree.isSubtree(tree.root1, tree.root2))    //if true then execute below
statement
            System.out.println("Tree 2 is subtree of Tree 1 ");
        else
            System.out.println("Tree 2 is not a subtree of Tree 1");
    }
}

```

Program 4: Deletion Operation for tow string

```

package ishwarchavan.com;

public class DeleteOperation {    //class created

    static int minDel(String s1, String s2){
        int i = 0;
        while (i < Math.min(s1.length(), s2.length())) {    // Iterate in the strings

            if (s1.charAt(i) != s2.charAt(i)){    //condition checking
                break;
            }
            i++;
        }

        int ans = ((s1.length() - i) + (s2.length() - i));    // Return the result
        return ans;
    }

    public static void main(String[] args){    //main program created
        String s1 = "ishwar";
        String s2 = "ishwarchavan";

        System.out.println(minDel(s1, s2));    //function calling
    }
}

```

