```java
import java.util.*;
public class CharCount {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		String str="java";
		Map<Character,Integer> map=new HashMap<>();
		for(Character c:str.toCharArray()){
			if(map.containsKey(c)){
				map.put(c, map.get(c)+1);
			}
			else{
				map.put(c, 1);
			}
		}
		System.out.println(map);

	}

}

//{a=2, v=1, j=1}
```

```java
import java.util.*;
public class palindrome {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		String str="nitin";
		int start=0;
		int end=str.length()-1;
		boolean isPalin=true;
		while(start<end){
			if(str.charAt(start)!=str.charAt(end)){
				isPalin=false;
				break;
			}
			start++;
			end--;
		}
		if(isPalin){
			System.out.print("yes");
```

```
            }
            else{
                  System.out.print("no");
            }
      }

}
//yes


import java.util.HashMap;
import java.util.*;

public class FreqOfEachChar {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            String str="java";
            Map<Character,Integer> map=new LinkedHashMap<>();

            for(int i=0;i<str.length();i++){
                  Character c=str.charAt(i);
                  if(map.containsKey(c)){
                        map.put(c, map.get(c)+1);
                  }
                  else{
                        map.put(c, 1);
                  }
            }
            System.out.println(map);
      }

}
//{j=1, a=2, v=1}--->maintain order of occurrence



import java.util.*;
public class CountOfWords {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            String str="java  programming  questions";
```

```
            //System.out.println(str.split("\\s+").length);//sapce tab
continues space 1 count

            System.out.println(countWord(str));
      }

      static int countWord(String str){
            int wordCount=1;

            for(int i=0;i<str.length();i++){
                  if(str.charAt(i)==' '  && str.charAt(i+1)!=' ' &&
i<str.length()-1){
                        wordCount++;
                  }
            }
            return wordCount;

      }

}
//3



import java.util.HashMap;
import java.util.Map;
import java.util.*;

public class DuplicateCharacter {


      static void printDuplicate(String str){


            Map<Character,Integer> map=new HashMap<>();
            for(Character c:str.toCharArray()){
                  if(map.containsKey(c)){
                        map.put(c, map.get(c)+1);
                  }
                  else{
                        map.put(c, 1);
                  }
            }
            Set<Map.Entry<Character,Integer>> se=map.entrySet();
            for(Map.Entry<Character,Integer> entry:se){
```

```
                if(entry.getValue()>1){
                        System.out.print(entry.getKey());
                        //System.exit(0);
                }
        }
    }
    static void printDuplicateCharSet(String str){
        Set<Character> st=new HashSet<>();
        for(int i=0;i<str.length();i++){
                Character ch=str.charAt(i);
                if(st.contains(ch)){
                        System.out.print(ch+" ");
                }
                else{
                        st.add(ch);
                }
        }
    }


    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String str="programming";
        //printDuplicate(str);
        printDuplicateCharSet(str);
    }

}
//r m g


import java.util.*;
public class ReverseEachWords {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String str="java programming";
        String[] words=str.split(" ");
        String revString="";

//          for(int i= words.length-1;i>=0;i--){
//              revString+=words[i]+" ";
//          }
//          System.out.print(revString);
```

```
            for(int i=0;i<words.length;i++){
                  String word=words[i];
                  String revWord="";
                  for(int j= word.length()-1;j>=0;j--){
                        revWord+=word.charAt(j);
                  }
                  revString+=revWord+" ";
            }
            System.out.print(revString);

      }

}
//programming java
//avaj gnimmargorp


import java.util.HashMap;
import java.util.Map;
import java.util.*;

public class NotReapeatingChar {

      static void printDuplicate(String str){


                  Map<Character,Integer> map=new   LinkedHashMap<>();
                  for(Character c:str.toCharArray()){
                        if(map.containsKey(c)){
                              map.put(c, map.get(c)+1);
                        }
                        else{
                              map.put(c, 1);
                        }
                  }
                  Set<Map.Entry<Character,Integer>> se=map.entrySet();
                  for(Map.Entry<Character,Integer> entry:se){
                        if(entry.getValue()==1){
                              System.out.print(entry.getKey());
                              System.exit(0);
                        }
                  }
            }

      public static void main(String[] args) {
```

```
            // TODO Auto-generated method stub
            String str="java";
            printDuplicate(str);

      }

}
//j



import java.util.*;
public class LongestCommonPrefix {
      //["cat","cable","camera"]

      public static String findLongestPrefix(String[] strs){
            String lcp=strs[0];//cat

            for(int i=0;i<strs.length;i++){
                  String currWord=strs[i];//cable
                  int j=0;

                  while(j<currWord.length() && j<lcp.length() &&
                              currWord.charAt(j)==lcp.charAt(j)){
                        j++;
                  }

                  if(j==0) return "";

                  lcp=currWord.substring(0,j);//ca

            }
            return lcp;

      }

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            String[] strs={"cat","cable","camera"};
            //String[] strs={"rat","cat","ghf"};
            System.out.println(findLongestPrefix(strs));

      }

}
```

```
//ca



import java.util.*;
public class LongestSubstring {

	public static int lenOfLongestSubstring(String s){
		int maxCount=0;
		int i=0,j=0;                              //i  |
		int l=s.length();                         // a b c a b c b
		Set<Character> st=new HashSet<>();        //     j
                                                  //{a, b, c ,a
		while(i<l && j<l){
			if(!st.contains(s.charAt(j))){
				st.add(s.charAt(j));
				j++;//unique ele move forward
				maxCount=Math.max(maxCount, j-i);  //3
			}else{
				st.remove(s.charAt(i));
				i++;
			}
		}

		return maxCount;

	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		String s="abcabcb";
		System.out.println(lenOfLongestSubstring(s));
	}

}
//3



import java.util.*;
public class Anagram {
    public static boolean isAnagram(String s1,String s2){

//    if(s1.length()!=s2.length()) return false;
//
//    char[] a1=s1.toCharArray();
```

```
//    char[] a2=s2.toCharArray();
//    Arrays.sort(a1);
//    Arrays.sort(a2);
//
//    return Arrays.equals(a1, a2);

      int[] countArr=new int[26];
      for(int i=0;i<s1.length();i++){
            int index=(int)s1.charAt(i)-97;
            countArr[index]++;

            int ind=(int)s2.charAt(i)-97;
            countArr[ind]--;
      }
      for(int j=0;j<countArr.length;j++){
            if(countArr[j]!=0) return false;
      }
      return true;

    }
      public static void main(String[] args) {
            // TODO Auto-generated method stub
            String s1="car";
            String s2="rac";
            boolean res=isAnagram(s1,s2);

            if(res)
              System.out.println("yes");
            else
              System.out.println("no");

      }

}

//yes




import java.util.*;
public class RemoveAdjDuplicates {

      public static String removeAdjDuplicates(String str){
            Stack<Character> st=new Stack<>();
```

```
            StringBuilder sb=new StringBuilder();

            for(int i=0;i<str.length();i++){
                  if(!st.isEmpty() && st.peek()==str.charAt(i)){
                        st.pop();
                  }
                  else{
                        st.push(str.charAt(i));
                  }
            }
            for(Character c:st){
                  sb.append(c);
            }
            return sb.toString();

      }
      public static void main(String[] args) {
            // TODO Auto-generated method stub
            String str="abbaca";
            System.out.println(removeAdjDuplicates(str));
      }

}
//ca



public class IsSubsequence {

      public static boolean isSubsequence(String s1,String s2){

            int i=0,j=0;

            while(i<s1.length() && j<s2.length()){

                  if(s1.charAt(i)==s2.charAt(j)) i++; // equal i inc

                  if(s1.length()==i) return true;

                  j++;
            }
            return false;

      }
```

```
	public static void main(String[] args) {
		// TODO Auto-generated method stub
	   String s1="ade";
	   String s2="adcdefgh";
	   boolean res=isSubsequence(s1,s2);
	   if(res)
	      System.out.print("yes");
	   else
	      System.out.print("no");
	}

}
//yes
```