

1A. ****filename:lab1A.java

```
/* Given a number, check if the number (N) can be written as the form (k+1)*k.
Write a java program to print those numbers in the given range.

Input Format:
-----
Line-1: An integer, N

Output Format:
-----
List of integers

Sample Input-1:
-----
10
Sample Output-1:
-----
2
6

Sample Input-2:
-----
15
Sample Output-2:
-----
2
6
12

*/
import java.util.*;
public class lab1A{
    public static void main(String args[]){
        Scanner sc= new Scanner(System.in);
        int n=sc.nextInt();
        for(int i=2;i<n;++i){
            if(i*(i-1) <= n){
                System.out.println(i*(i-1));
            }
        }
    }
}
```

1B. ****filename:lab1B.java

```
/* Write a java program to check whether the given number is gapful or not.
```

A number is gapful if it is at least 3 digits long and is divisible by the number formed by stringing the first and last numbers together.

The smallest number that fits this description is 100.

First digit is 1, last digit is 0, forming 10, which is a factor of 100.

Therefore, 100 is gapful.

Input Format:

Line-1: An integer, N

Output Format:

Line-1: Boolean value True or False

Sample Input-1:

100

Sample Output-1:

True

Sample Input-2:

25

Sample Output-2:

False

```
*/
import java.util.*;
class lab1B{
    public static void main (String[] args) {
        Scanner sc = new Scanner(System.in);
        String n=sc.next();
        String
r=Character.toString(n.charAt(0))+Character.toString(n.charAt(n.length()-1));

        if(n.length()>2){
            if(Integer.parseInt(n)%Integer.parseInt(r)==0){
                System.out.println("True");
            }
        }
        else{
            System.out.println("False");
        }
    }
}
```

```
}  
}
```

1C. ****filename:lab1C.java

```
/*Cricketer's Pension Continuing our journey in mastering the conditional  
statements & our  
interest with cricket.  
Let us help the Indian cricket's governing body (BCCI) to automate its plan of  
allotting pensions  
to former players.  
The rules are given below:  
If a player has played more than 10 test matches and 100 ODI's he receives  
Rs.50,000.  
If a player has played more than 10 test matches he receives Rs.25,000.  
If a player has played more than 100 ODI's he receives Rs.15,000.  
If a player has played for India he receives Rs.10,000.  
The amount is incremented by 1/4th for every 'man of the match' award.  
If a player has not played for India but played IPL he receives an amount of  
Rs.8,000.  
If a player has not played for India nor IPL he receives an amount of  
Rs.7,000.*/  
import java.util.*;  
public class lab1C  
{  
    public static void main(String[] args) {  
        Scanner sc=new Scanner(System.in);  
        System.out.println("has the player played for india[y/n]?");  
        String ch=sc.next();  
        System.out.println("enter the no of test matches he has played:");  
        int test_matches=sc.nextInt();  
        System.out.println("enter the no of ODI matches he has played:");  
        int ODI_matches=sc.nextInt();  
        System.out.println("no of man of the matches he has won:");  
        int man_of_the_matches=sc.nextInt();  
        float amount=0.0f;  
        if(test_matches>10 &&ODI_matches>100){  
            amount+=50000;  
        }  
        else if(test_matches>10){  
            amount+=25000;  
        }  
        else if(ODI_matches>100){  
            amount+=15000;  
        }  
        else if(ch.equals('y')){  
            amount+=10000;  
        }  
    }  
}
```

```

amount=amount+(amount*(man_of_the_matches*0.25f));
System.out.println(amount);

}
}

```

3A. ****filename:lab3A.java

```

/* Given an array arr[] of N integers,
the task is to find the maximum difference between any two elements of the
array.

```

Input Format:

Line-1: An integer N, represents array elements

Line 2: Array elements

Output Format:

Line-1: An integer, result

Sample Input-1:

5

10 20 30 40 50

Sample Output-1:

40

Sample Input-2:

5

-1 2 0 -5 10

Sample Output-2:

15

*/

```
import java.util.*;
```

```
class lab3A{
```

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

```
        Scanner sc=new Scanner(System.in);
```

```
        int n=sc.nextInt();
```

```
        int[] arr=new int[n];
```

```
        for(int i=0;i<n;i++)
```

```
        {
```

```
            arr[i]=sc.nextInt();
```

```
        }
```

```

        int maximum=0;
        int minimum=99999;
        for(int i=0;i<n;i++)
        {
            if(arr[i]>maximum)
            {
                maximum=arr[i];
            }
            if(arr[i]<minimum)
            {
                minimum=arr[i];
            }
        }
        int diff=maximum-minimum;
        System.out.println(diff);
    }
}

```

3B. ****filename:lab3B.java

```

/* Write a Java program to fill the below pattern into a square matrix:

The matrix has to be filled with numbers starting from 1.
It has to start fill first row, last column, last row (reverse), first column
(reverse) and so on.

Input Format:
-----
Line-1: An integer N, represents array elements

Output Format:
-----
A square matrix, result

Sample Input-1:
-----
5
Sample Output-1:
-----
1 2 3 4 5
16 17 18 19 6
15 24 25 20 7
14 23 22 21 8
13 12 11 10 9

*/
import java.util.*;

```

```

public class lab3B
{
    static void bam(int size)
    {
        int r = 0, c = 0;
        int boundary = size - 1;
        int sizeLeft = size - 1;
        int flag = 1;
        char move = 'r';
        int[][] matrix = new int [size][size];
        for (int i = 1; i < size * size + 1; i++)
        {
            matrix[r][c] = i;

            switch (move)
            {
                case 'r':
                    c += 1;
                    break;
                case 'l':
                    c -= 1;
                    break;
                case 'u':
                    r -= 1;
                    break;
                case 'd':
                    r += 1;
                    break;
            }

            if (i == boundary)
            {
                boundary = boundary + sizeLeft;
                if (flag != 2)
                {
                    flag = 2;
                }
                else
                {
                    flag = 1;
                    sizeLeft -= 1;
                }

                switch (move)
                {

```

```

        case 'r':
            move = 'd';
            break;
        case 'd':
            move = 'l';
            break;
        case 'l':
            move = 'u';
            break;

        case 'u':
            move = 'r';
            break;
    }
}

for (r = 0; r < size; r++)
{

    for (c = 0; c < size; c++)
    {
        int n = matrix[r][c];
        if(n < 10)
            System.out.print(n + " ");
        else
            System.out.print(n + " ");
    }
    System.out.println();
}

}

public static void main(String args[])
{
    Scanner sc =new Scanner(System.in);
    int size = sc.nextInt();
    bam(size);
}
}

```

3C. ****filename:lab3C.java

```

/* Given n non-negative integers representing an elevation map where the width
of each bar is 1,
compute how much water it is able to trap after raining.

```

Input Format:

Line-1: An integer N, represents array elements

Line 2: Array elements

Output Format:

Line-1: An integer, result

Sample Input-1:

6

3 0 0 2 0 4

Sample Output-1:

10

NOTE : Refer Hint for explanation.

```
*/
import java.util.*;
class lab3C{
public static int element(int[] arr, int n)
{
int a = 0;

for(int i = 1; i< n - 1; i++)
{
int left = arr[i];
for(int j = 0; j <i; j++)
{
left = Math.max(left, arr[j]);
}
int right = arr[i];
for(int j = i + 1; j < n; j++)
{
right = Math.max(right, arr[j]);
}
a += Math.min(left, right) - arr[i];
}
return a;
}
public static void main(String[] args)
{
Scanner sc =new Scanner(System.in);
int a=sc.nextInt();
int[] arr = new int[a];
for(int i=0;i<a;i++){
arr[i]=sc.nextInt();
}
```



```

}
System.out.print(element(arr,a));
}
}

```

4A. ****filename:lab4A.java

```

/* Given two strings text1 and text2, return the length of their longest
common subsequence.

```

A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

(eg, "ace" is a subsequence of "abcde" while "aec" is not).

A common subsequence of two strings is a subsequence that is common to both strings.

NOTE : If there is no common subsequence, return 0.

Input Format:

Line-1: A string, represents text1

Line 2: A string, represents text2

Output Format:

Line-1: An integer, result

Sample Input-1:

abcdefghijkl

caeghiklp

Sample Output-1:

6

*/

```

import java.util.*;

```

```

class lab4A

```

```

{

```

```

    public static int element(String t1, String t2)

```

```

    {

```

```

        int [][]matrix=new int[t1.length()+1][t2.length()+1];

```

```

        for(int i=1;i<=t1.length();i++)

```

```

        {

```

```

            for(int j=1;j<=t2.length();j++)

```

```

            {

```

```

if(t1.charAt(i-1)==t2.charAt(j-1)){
matrix[i][j]=matrix[i-1][j-1]+1;
}
else
{
matrix[i][j]=Math.max(matrix[i-1][j],matrix[i][j-1]);
}
}
}
return matrix[t1.length()][t2.length()];
}
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
System.out.println(element(sc.next(),sc.next()));
}
}

```

4B. ****filename:lab4B.java

```

/* Given two strings s1 and s2, your task is to merge those strings to form a
new merged string.

```

A merge operation on two strings is described as follows:
Append alternating characters from s1 and s2, respectively, to mergedString.
Once all of the characters in one of the strings have been merged,
append the remaining characters in the other string to mergedString.

Input Format:

Line-1: A string, represents text1

Line 2: A string, represents text2

Output Format:

Line-1: A string, result

Sample Input-1:

abc

def

Sample Output-1:

adbecf

Sample Input-2:

abcdef

```

pqr
Sample Output-2:
-----
apbqcrdef

*/
import java.util.*;
public class lab4B{
    public static String element(String s1, String s2) {
        StringBuilder s = new StringBuilder();
        int i1 = 0, i2 = 0;
        while (i1 < s1.length() && i2 < s2.length()) {
            s.append(s1.charAt(i1));
            s.append(s2.charAt(i2));
            i1++;
            i2++;}
        while (i1 < s1.length()) {
            s.append(s1.charAt(i1));
            i1++; }
        while (i2 < s2.length()) {
            s.append(s2.charAt(i2));
            i2++;}
        return s.toString();
    }
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println(element(sc.next(),sc.next()));
    }
}

```

4C. ****filename:lab4C.java

```

/*You are given a string S of alphabet characters and the task is to find its
matching decimal
representation as on the shown keypad. Output the decimal representation
corresponding to the
string.
For ex: if you are given "amazon" then its corresponding decimal
representation will be 262966.*/
import java.io.*;
import java.util.*;
class lab4C
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        String str;

```

```
str=sc.next();
char[] ch = new char[str.length()];
for (int i = 0; i<str.length(); i++)
{
ch[i] = str.charAt(i);
switch (ch[i])
{
case 'a':
case 'b':
case 'c':
System.out.print("2");
break;
case 'd':
case 'e':
case 'f':
System.out.print("3");
break;
case 'g':
case 'h':
case 'i':
System.out.print("4");
break;
case 'j':
case 'k':
case 'l':
System.out.print("5");
break;
case 'm':
case 'n':
case 'o':
System.out.print("6");
break;
case 'p':
case 'q':
case 'r':
case 's':
System.out.print("7");
break;
case 't':
case 'u':
case 'v':
System.out.print("8");
break;
case 'w':
case 'x':
case 'y':
case 'z':
System.out.print("9");
```

```

break;
default:
System.exit(0);
break;
}} }}

```

5.*****filename:MyClass.java

```

/*Define a MyRectangle class with four public data members representing the x-
and ycoordinates of the bottom-left vertex and top-right vertex of a
rectangle, whose sides are
parallel to the x- or y-axis.
For example, the statement new MyRectangle(20,80,30,90) creates a rectangle
with bottomleft vertex at position (20,80), and top-right vertex at (30,90).
(a) Write an area() method, which computes the area of a rectangle.
(b) Write the overlap(MyRectanglerect) method. This method returns a rectangle
which is
the overlapped region of two rectangles. In the event that there is no
overlap, it should
return a rectangle with both bottom-left vertex and top-right vertex at
position (0,0).
(c) Using the overlap(MyRectanglerect) method written above, write the
overlapAll(MyRectangle[] rectangles) method which returns the overlapped
region of all
the rectangles in the array. You may assume that there is at least one element
in the array.
Your method should be efficient in that the moment it finds that the
overlapped region is
empty, it should return a rectangle with both vertices at (0,0) immediately.
(d) Write MySqaure.java, MySquare extends MyRectangle.
A square is defined by its bottom-left vertex and the length of its side.
Complete the super( .
. . ) statement in the constructor.
(e)Below is output of MySquare.java program when the user enters: 10 30 5.
Override the toString() method in MyRectangle in order to get such output.*/
import java.util.*;
class Myrect{
    public int a;
    public int b;
    public int c;
    public int d;
    public Myrect(int x1,int x2,int y1,int y2){
        a=x1;
        b=y1;
        c=x2;
        d=y2;

    }
}

```

```

    public void area(){
        System.out.println(Math.abs((a-c)*(b-d)));
    }
    public void over(Myrect f){
        int x1=Math.max(a,f.a);
        int y1=Math.max(b,f.b);
        int x2=Math.min(c,f.c);
        int y2=Math.min(d,f.d);
        if(x1>x2 || y1>y2){
            Myrect m1=new Myrect(0,0,0,0);

        }
        else{
            Myrect m1=new Myrect(x1,x2,y1,y2);

        }

    }
}
class Mysquare{
    int g;
    int ui;
    int lo;
    public Mysquare(int a,int b,int c){
        g=a;
        ui=b;
        lo=c;
        System.out.println(c*c);

    }

}
public class MyClass{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        int r=sc.nextInt();
        int p=sc.nextInt();
        int q=sc.nextInt();
        Myrect m=new Myrect(r,p,q,r);
        Myrect m2=new Myrect(r,p,q,q);
        Mysquare ymir=new Mysquare(r,p,q);
        m.over(m2);

    }
}

```

2. ***filename:lab2.java

```
// Write a Resistor class containing the parameterized constructor, which
// takes in three
// strings representing the three band colours. Write the methods to calculate
// and set the
// resistance for the resistor.
import java.util.*;
class Resistor{
    HashMap<String,Integer> color;
    // data members
    private String band1;
    private String band2;
    private String band3;
    private double value; // Resistor value (Ohms)
    // default constructor
    public Resistor()
    {
        band1=band2=band3="";
        value=0.0;
    }
    // Constructor taking three strings as parameters
    // and calculating the resistance
    public Resistor(String b1, String b2, String b3)
    {
        // *** to be completed ***
        band1=b1;
        band2=b2;
        band3=b3;
        color=new HashMap<>();
        color.put("black",0);
        color.put("brown",1);
        color.put("red",2);
        color.put("orange",3);
        color.put("yellow",4);
        color.put("green",5);
        color.put("blue",6);
        color.put("violet",7);
        color.put("grey",8);
        color.put("white",9);
    }
    public String getResistance(){
        int firstdig=color.get(band1);
        int seconddig=color.get(band2);
        int thirddig=(int)Math.pow(10,color.get(band3));
        String value1=Integer.toString(firstdig)+Integer.toString(seconddig);

        value=Integer.parseInt(value1)*thirddig;
        value=value/1000;
    }
}
```

```

    value1=value+"Kilo_ohms";
    return value1;
}
}
public class lab2{
public static void main(String args[]){
Resistor r1=new Resistor("red","black","orange");
System.out.println(r1.getResistance());
}
}
}

```

6a. ****filename:lab6A.java

```

// 6. a. Write a java program that loads names and phone numbers from the text
// file into Hash
// Table where data is organized as one line per record and each field in
// record are separated by a
// tab(\t). It takes a name or phone number as input and prints the
// corresponding other value from
// hash table.
import java.util.*;
import java.io.*;
public class lab6A
{ public static void main(String[] args) throws Exception
{ Scanner sc = new Scanner(System.in);
  FileReader fr = new FileReader("phone.txt");
  BufferedReader br=new BufferedReader(fr);
  Hashtable ht=new Hashtable();
  String line,name, mobile;
  while((line=br.readLine())!=null)
  { String s[] = line.split("\\s+");
    ht.put(s[0],s[1]); }
  System.out.println(ht);
  System.out.print("Enter name: ");
  name = sc.next();
  if (ht.containsKey(name))
  { System.out.println("Mobile number is " + ht.get(name)); }
  else { System.out.println("Not Found"); }
  System.out.print("Enter mobile number: ");
  mobile = sc.next();
  if (ht.containsValue(mobile))
  {
  Set<Map.Entry> s1=ht.entrySet();
  for (Map.Entry e : s1)
  { if (mobile.equals(e.getValue()))
  { System.out.println("Name is " + e.getKey()); } } }
  else { System.out.println("Not Found"); }
  br.close(); } }

```


OUTPUT:**SAMPLE INPUT:**

FIRST CREATE A FILE PHONE.TXTAN THEN GIVE INPUT AS:

MAHESH 9116161

PAVANI 6166266

BINDU 795446

ARJUN 630245

NIKITHA 548932

6b. ****filename:lab6B.java

```
// Write a Java program to count the votes from the logfile, throwing out
duplicate votes
// using the rules above. Display the votes in a table, as shown below for the
example:
// PHP 2
// C# 1
// Prolog 1
import java.util.*;
import java.io.*;
import java.util.concurrent.CopyOnWriteArrayList;
class Data{
String plang,ip,timestamp;
void setData( String plang, String ip,String timestamp)
{
this.plang=plang;
this.ip=ip;
this.timestamp=timestamp;
}
String getPlang()
{
return this.plang;
}
String getIP()
{
return this.ip;
}
String getTimestamp()
{
return this.timestamp;
}
}
public class lab6B
{
```

```

public static void main(String[] args) throws Exception
{
    Scanner sc = new Scanner(System.in);
    FileReader fr = new FileReader("phone2.txt");
    BufferedReader br=new BufferedReader(fr);
    String line;
    String[] data;
    List<Data>al = new CopyOnWriteArrayList<Data>();
    int a =Integer.parseInt(br.readLine());
    for(int i=0;i<a;i++){
        Data d=new Data();
        String r=br.readLine();
        String s=br.readLine();
        String t=br.readLine();
        d.setData(r,s,t);
        al.add(d);
    }
    String[] ar=new String[]{"PHP","C#","Prolog"};
    int[] ar1=new int[]{1,1,1};
    String[] ar2=new
    String[]{"137.229.156.12","137.229.156.18","156.213.38.31","128.120.56.214"};
    int[] ar3=new int[]{0,0,0,0};
    for(Data i : al){
        String u=i.getPlang();
        String w=i.getIP();
        String x=i.getTimestamp();
        int k =0;
        for(int j=0;j<ar.length;j++){
            if(ar[j]==u){
                k=j;
                break;
            }
        }
        int z=0;
        for(int j=0;j<ar2.length;j++){
            if(ar2[j]==w){
                z=j;
                break;
            }
        }
        if(ar3[z]==0){
            ar1[k]++;
            ar3[z]=1;
        }
    }
    for(int i=0;i<ar.length;i++){
        System.out.print(ar[i]);
        System.out.print(" ");
    }
}

```

```

System.out.print(ar1[i]);
System.out.println();
}
for (Data d: a1) {
if (d.getPlang().equals(d.plang)) {
if (d.getIP().equals(d.ip))
{
a1.remove(d);
}
}
}
} }

```

OUTPUT:

SAMPLE INPUT:

FIRST CREATE A FILE phone2.txt AND THEN GIVE INPUT AS:

6

PHP

137.229.156.12

1000002

C#

137.229.156.18

1000005

PHP

137.229.156.12

1000006

Prolog

156.213.38.31

1000010

PHP

128.120.56.214

1000020

PHP

137.229.156.12

1000022

OUTPUT:

PHP 2

C# 1

Prolog 1

7a. ****filename:lab7A.java

```
// a.Write a Java Class to implement a method Addition() that returns a new
// Array where each
// array element at the index k corresponds to the sum of elements of the
// array (src) starting at
// index 0 and including element at the index 'k'. For example, for array
// [2,3,5], the method
// will return array [2,5,10]. For an array of size '0' or a null parameter,
// the method will throw
// exception IllegalArgumentExceptionWith the message "Invalid Argument".
import java.util.Arrays;
import java.util.Scanner;
class lab7A
{
    public static void main(String[] args)
    {
        System.out.println("Enter Array Size: ");
        Scanner sc=new Scanner (System.in);
        int n=sc.nextInt();
        int arr[];
        try
        {
            if (n>0)
            arr=new int[n];
            else
            throw new IllegalArgumentException() ;
            System.out.println("Enter Array Elements: ");
            for (int i=0;i<n ;i++ )
            {
                arr[i]=sc.nextInt();
            }
            lab7A rs=new lab7A();
            int resultarr[]=rs.addition(arr);
            System.out.println(Arrays.toString(resultarr));
        }
        catch (IllegalArgumentException e)
        {
            System.out.println("Invalid Argument");
        }
    }
}
```

```

}
}
int[] addition(int[] a) {
int sum = 0;
for (int i = 0; i<a.length; i++) {
a[i] += sum;

sum = a[i];
}
return a;
}
}

```

7b. ****filename:lab7B.java

```

// b. Write a Java Code to implement a multithreaded version of FizzBuzz with
four threads.
// If the number is divisible by 3, output "fizz".
// If the number is divisible by 5, output "buzz".
// If the number is divisible by both 3 and 5, output "fizzbuzz".
// If the number is not divisible by both 3 and 5 print the number
// For instance if n is 15 , we will have the output as “ 1, 2, fizz, 4, buzz,
fizz, 7, 8, fizz,buzz,
// 11, fizz, 13, 14, fizzbuzz “
// 1. Thread A will call fizz() to check for divisibility of 3 and outputs
fizz .
// 2. Thread B will call buzz() to check for divisibility of 5 and outputs
buzz .
// 3. Thread C will call fizzbuzz() to check for divisibility of 3 and 5 and
outputs fizzbuzz .
// 4. Thread D will call number() which should only output the numbers.
import java.util.Scanner;
class lab7B

{
public static void main(String[] args) throws Exception
{

Scanner sc=new Scanner(System.in);
System.out.println("Enter n value: ");
int n=sc.nextInt();
for(int i=1;i<=n ;i++)
{
if (i%15 == 0)
{
C t1=new C();
t1.start();
t1.join();

```

```

}
    else if (i%5==0)
    {
        B t2=new B();
        t2.start();
        t2.join();
    }
    else if (i%3 == 0)
    {
        A t3=new A();
        t3.start();
        t3.join();
    }
    else
        System.out.print(i+", ");
}
}
}
class A extends Thread
{
    public void run()
    {
        System.out.print("fizz, ");
    }
}
class B extends Thread
{
    public void run()
    {
        System.out.print("buzz" +", ");
    }
}
class C extends Thread
{
    public void run()
    {
        System.out.print("fizzbuzz" +", ");
    }
}
}

```

8a. ****filename:lab8A.java

```

// 8.a.Write a java program to store the employee details in an ArrayList and
display the
// employee details in ascending order of their experience. Create 'Employee'
class with two
// instance variables Employee name and Employee experience (no. of years).
import java.util.*;

```

```

class lab8A
{
public static void main(String[] args)
{
ArrayList<Employee>ar = new ArrayList<Employee>();
ar.add(new Employee("John",5));
ar.add(new Employee("Nic",3));
ar.add(new Employee("Roma",9));
System.out.println("Unsorted Emp List");
for (int i = 0; i<ar.size(); i++)
System.out.println(ar.get(i).toString());
Collections.sort(ar, new Sortbyexp());
System.out.println("Sorted by Experience");
    for (int i = 0; i<ar.size(); i++)
System.out.println(ar.get(i).toString());
}
}
class Employee
{
String empname;
int exp;
Employee(String empname, int exp)
{
this.empname= empname;
this.exp=exp;
}
public String toString()
{
    return this.empname + " " + this.exp ;
}
}
class Sortbyexp implements Comparator<Employee> {
    public int compare(Employee a, Employee b)
    {
        return a.exp - b.exp;
    }
}
}

```

8b. ****filename:lab8B.java

```

// b. Write a program to find the most common words in the list of words given
in sorted order
// based on occurrence from largest to smallest. If any of words are having
same occurrence then
// consider the smallest character order comes first.
// Input format: First line contains the list of words and next line contains
a number (k) which
// represent the top most words to display.

```

```
// Output format: display the k top most words.
import java.util.*;
class lab8B{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        String s=sc.nextLine();
        String arr[]=s.split(" ");
        Map <String,Integer>ch=new LinkedHashMap<String,Integer>();
        for(int i=0;i<arr.length;i++){
            String x=arr[i];
            if(ch.containsKey(x)){
                int c=ch.get(x);
                c++;
                ch.put(x,c);
            }
            else{
                ch.put(x,1);
            }
        }
        ArrayList<Integer> al=new ArrayList<Integer>();
        Set<String> ss=ch.keySet();
        for(String i:ss){
            if(al.contains(ch.get(i))){}
            else{
                al.add(ch.get(i));}
        }

        Collections.sort(al);
        for(int i=al.size()-1;i>=0;i--){
            int temp= al.get(i);
            ArrayList<String>tal=new ArrayList<>();
            for(Map.Entry<String, Integer> l1 : ch.entrySet())
            {
                if(l1.getValue() == temp)
                    tal.add(l1.getKey());
            }
            Collections.sort(tal);
            for( String s1:tal)
                System.out.print(s1+" ");
        }
    }
}
```

Expected output: lets do one to do how to do

do to how lets one

9a. ****filename:lab9A.java

```
// 9. a. Write a java Program to write a method fCount which takes a string as
a parameter.
```



```

// The Method fCount should return the Map which has the frequency count of
the given
// word. For example if the string passed is "hello" the map should return {h-
1,e-1,l-2,o-1}.
// The order of the characters should be same as in the string.
import java.util.*;
class lab9A
{
public static void main(String args[])
{
Scanner input = new Scanner(System.in);
System.out.print("Enter a string : ");
String str = input.next();
Map<Character, Integer> result = new HashMap<Character, Integer>();
result = fCount(str);
for(int i = 0; i<str.length(); i++)
{
if(result.get(str.charAt(i)) != 0)
{
System.out.print(str.charAt(i)+"-"+result.get(str.charAt(i)) + " ");
result.put(str.charAt(i), 0);
}
}
input.close();
}
public static Map<Character, Integer>fCount(String str)
{
Map<Character, Integer> map = new HashMap<Character, Integer>();
for(int i = 0; i<str.length(); i++)
{
if(map.containsKey(str.charAt(i)))
{
map.put(str.charAt(i), map.get(str.charAt(i)) + 1);
}
else
{
map.put(str.charAt(i), 1);
}
}
return map;
}
}

```

9b. ****filename:lab9B.java

```

// In the example above, these two HashMaps have two key/value pairs in
common, namely:

```

```

// "Mary"/"Ecstatic" and "Bob"/"Happy". Note that although the key "Felix" is
// in both
// HashMaps, the associated value with this key is different in the two maps
// (hence this does
// not count as a key/value pair that is common to both HashMaps). Similarly,
// just having the
// same value without the same key (such as the value "Fine" which is mapped
// to by different
// keys in the two different HashMaps) would also not count as a common
// key/value pair
// between the two HashMaps.
import java.util.*;
class lab9B
{
    public static void main(String args[])
    {
        Scanner input = new Scanner(System.in);
        HashMap<String, String>mapOne = new HashMap<String, String>();
        HashMap<String, String>mapTwo = new HashMap<String, String>();
        System.out.print("Enter how many pairs in HashMap 1 : ");
        int mapOneSize = input.nextInt();
        input.nextLine();
        System.out.println("Enter your " + mapOneSize + " pair of String ");
        for(int i = 0; i<mapOneSize; i++)
        {
            System.out.println("Enter Pair " + (i+1));
            String pairs[] = input.nextLine().split(" ");
            mapOne.put(pairs[0],pairs[1]);
        }
        System.out.print("Enter how many pairs in HashMap 2 : ");
        int mapTwoSize = input.nextInt();
        input.nextLine();
        System.out.println("Enter your " + mapTwoSize + " pair of String ");
        for(int i = 0; i<mapTwoSize; i++)
        {
            System.out.println("Enter Pair " + (i+1));
            String pairs[] = input.nextLine().split(" ");
            mapTwo.put(pairs[0],pairs[1]);
        }
        int result = commonKeyValuePairs(mapOne,mapTwo);
        System.out.println("The number of key/value pairs are : " + result);
        input.close();
    }
    public static int commonKeyValuePairs(HashMap<String, String> map1 ,
    HashMap<String, String> map2)
    {
        int count = 0;
        for(Map.Entry m1:map1.entrySet())

```

```

{
for(Map.Entry m2:map2.entrySet())
{
if(m1.getKey().equals(m2.getKey())&& m1.getValue().equals(m2.getValue()))
{
count++;
}
}
}
return count;
}
}

```

EXPECTED OUTPUT Enter how many pairs in HashMap 1 : 5 Enter your 5 pair of String Enter Pair 1 Alice Healthy Enter Pair 2 Mary Ecstatic Enter Pair 3 Bob Happy Enter Pair 4 Chuck Fine Enter Pair 5 Felix Sick Enter how many pairs in HashMap 2 : 5 Enter your 5 pair of String Enter Pair 1 Mary Ecstatic Enter Pair 2 Felix Healthy Enter Pair 3 Ricardo Superb Enter Pair 4 Tam Fine Enter Pair 5 Bob Happy The number of key/value pairs are : 2