

Functions

Q1. Write a class to check whether a given number is Armstrong or not using a function name `int checkArmstrong(int num)`. Function should return a value 1 if number is Armstrong otherwise it return 0 if not.

```
public class ques1
{
    public static int isArmstrong(int number)
    {
        int orgNumber = number, sum = 0;
        while(number > 0)
        {
            int digit = number % 10;
            sum = sum + (int) Math.pow(digit, 3);
            number = number / 10;
        }
        if(orgNumber == sum)
            return 1;
        else
            return 0;
    }

    public static void main(int N)
    {
        if(isArmstrong(N) == 1)
            System.out.println("Number is Armstrong");
        else
            System.out.println("Number is not Armstrong");
    }
}
```

Q2. Write a class using a function `primeCheck(int num)` to check whether a given number is Prime or not. Function should return a value 1 if number is prime otherwise it return 0 if not.

```
public class ques2
```

```

{
    public static int primeCheck(int num)
    {
        int flag = 1;
        for(int i = 2;i<num;i++)
        {
            if(num%i == 0)
            {
                flag = 0;
                break;
            }
        }
        return(flag);
    }
}

```

Q3. Write a class using a function as follows: int num (int) the function accepts a number and finds whether it is even and divisible by 8 or not. It returns '1' if condition is satisfied otherwise '0'. Use a main program to pass the number by value to the function.

```

public class ques3
{
    public static int num(int num)
    {
        if(num%2 == 0 && num%8 == 0)
            return(1);
        else
            return(0);
    }
    public static void main(int N)
    {
        if(num(N) == 1)
            System.out.println ("Number is Divisible by 8 and Even");
        else

```

```

        System.out.println ("Number is not Divisible by 8 and Even");
    }
}

```

Q4. Write a class program 'series' with a function by using following specifications- int fact(int).

The function calculates and returns factorial of a number. Apply this function to find the sum of the following series in the main-function.

(Note : $n! = 1 \times 2 \times 3 \times \dots \times n$)

$S = 1 + \frac{1}{2!} + \frac{1}{4!} + \frac{1}{6!} + \dots + \frac{1}{10!}$

```

class Series
{
    public int fact(int N)
    {
        int f = 1;
        for(int i = 1;i <= N;i++)
            f = f*i;
        return(f);
    }
}

public class ques4
{
    public static void main()
    {
        Series s = new Series();
        double sum = 0;
        for(int i = 2;i <= 10;i += 2)
            sum = sum + (double)1/s.fact(i);
        System.out.println ("Sum S = "+sum);
    }
}

```

Q5. Write a menu driven program (using switch case) to find the areas of circle, square or rectangle. Use function overloading for each of the given shapes.

Q6. Develop a class "Array" with the following Specifications:

instant Variables- int a[20], k

Member functions-

public void input() To input integer members to array A and a number separately to variable k.

public void search() To find and print 'Search Successful' if number is found 'Search Unsuccessful' otherwise.

```
import java.io.*;
```

```
class Array
```

```
{
```

```
    int a[] = new int[20];
```

```
    int k;
```

```
    public void input() throws IOException
```

```
    {
```

```
        InputStreamReader isr = new InputStreamReader(System.in);
```

```
        BufferedReader x = new BufferedReader(isr);
```

```
        for(int i = 0;i<20;i++)
```

```
        {
```

```
            System.out.print ("Enter a number : ");
```

```
            a[i] = Integer.parseInt ( x.readLine());
```

```
        }
```

```
        System.out.print ("Enter a No. to Search : ");
```

```
        k = Integer.parseInt ( x.readLine());
```

```
    }
```

```
    public void search()
```

```
    {
```

```
        boolean flag = false;
```

```
        for(int i = 0;i<20;i++)
```

```
        {
```

```
            if(a[i] == k)
```

```

        flag = true;
    }
    if(flag == false)
        System.out.println ("Number not found in Array");
    else
        System.out.println ("Number found in Array");
    }
}

public class ques6
{
    public static void main() throws IOException
    {
        Array a = new Array();
        a.input();
        a.search();
    }
}

```

Q7. Define a subclass sub1 in which define a method f1() to calculate and print hcf (highest common factor) by division method of any two given numbers entered by user in the main class. Define another subclass sub2 in which define a method f2() to calculate and print the area and perimeter of a rectangle by using the required parameters accepted in main class. Define main class to input the parameters required in the above two methods and also call the two functions.

```

import java.io.*;

class sub1
{
    public void f1(int x,int y)
    {
        int n = (x<y)?x:y;
        int hcf = 1;
        for(int i = 1;i<= n;i++)
        {
            if(x%i == 0 && y%i == 0)a

```

```

        hcf = i;
    }
    System.out.println ("HCF = "+hcf);
}
}
class sub2
{
    public void f2(int l,int b)
    {
        System.out.println ("Length : "+l);
        System.out.println ("Breadth : "+b);
        int area = l*b;
        System.out.println ("Area : "+area);
        int perimeter = 2*(l+b);
        System.out.println ("Perimeter : "+perimeter);
    }
}
public class ques7
{
    public static void main() throws IOException
    {
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader x = new BufferedReader(isr);
        sub1 a = new sub1();
        sub2 b = new sub2();
        System.out.print ("Enter two Numbers : ");
        int N1 = Integer.parseInt ( x.readLine());
        int N2 = Integer.parseInt ( x.readLine());
        a.f1(N1,N2);
        System.out.print ("Enter Length : ");
        int L = Integer.parseInt ( x.readLine());
    }
}

```

```

        System.out.print ("Enter Breadth : ");

        int B = Integer.parseInt ( x.readLine());

        b.f2(L,B);

    }

}

```

Q8. Define a subclass str1 in which define a method rev1() to input a word and display it in reverse order. Define another method rev2() in the same subclass to input any 10 words and display the words having odd number of characters. Define main class to execute two functions.

```

import java.io.*;

class str1
{
    public void rev1() throws IOException
    {
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader x = new BufferedReader(isr);

        System.out.print ("Enter a Word : ");

        String n = x.readLine();

        int l = n.length();

        for(int i = l-1;i >= 0;i--)
        {
            System.out.print (n.charAt(i));
        }

        System.out.println ();
    }

    public void rev2() throws IOException
    {
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader x = new BufferedReader(isr);

        String n[] = new String[10];

        for(int i = 0;i<10;i++)
        {
            System.out.print ("Enter a Word : ");

```

```

        n[i] = x.readLine();
    }
    System.out.println ("Word having odd number of character");
    for(int i = 0;i<10;i++)
    {
        if((n[i].length())%2 == 1)
            System.out.println (n[i]);
    }
}

public class ques8
{
    public static void main() throws IOException
    {
        str1 a = new str1();
        a.rev1();
        a.rev2();
    }
}

```

Q9. Write a program to print the few lines of the following patter (number of lines is given by user).
Use one function for printing alphabets and another function to print the number.

A A A A A 1

B B B B 1 2

C C C 1 2 3

D D 1 2 3 4

E 1 2 3 4 5

```

class Pattern
{
    int N;
    public void alpha(int i)
    {

```



```

        char ch = (char)(64+i);
        for(int j = 1;j<= (N+1)-i;j++)
            System.out.print (ch+" ");
    }
    public void digit(int i)
    {
        for(int j = 1;j<= i;j++)
            System.out.print (j+" ");
    }
}

public class ques9
{
    public static void main(int N)
    {
        Pattern p = new Pattern();
        p.N = N;
        for(int i = 1;i<= N;i++)
        {
            p.alpha(i);
            p.digit(i);
            System.out.println ();
        }
    }
}

```

Q10. The number 151 is a prime palindrome, because it is both a prime number and a palindrome. Write a class that find all prime palindromes between two given numbers a and b. Accept the values for a and b from the user in function main(). Use two function in your class, boolean isPrime(int) for check number is prime or not, int isPalindrome(int) to check number is palindrome or not.

```

public class ques10
{
    public static boolean isPrime(int num)
    {

```

```

boolean flag = true;
for(int i = 2;i<num;i++)
{
    if(num%i == 0)
    {
        flag = false; break;
    }
}
return(flag);
}

public static int isPalindrom(int num)
{
    int reverseNumber = 0, orgNumber = num, digit;
    while(num>0)
    {
        digit = num%10;
        reverseNumber = reverseNumber*10 + digit;
        num = num/10;
    }
    if(orgNumber == reverseNumber)
        return(1);
    else
        return(0);
}

public static void main(int a, int b)
{
    for(int N = a;N<= b;N++)
    if(isPrime(N) && isPalindrom(N) == 1)
        System.out.println ("Number is "+N);
}
}

```

Q11. Create a class BubbleSort to arrange 20 integer type values in ascending order or descending order using the function "void bubbleSort(int n[],int so)". Here "so" stands for the sort order i.e. +1 for ascending and -1 for descending.

```
public class ques11
{
    public static void bubbleSort(int a[],int so)
    {
        int i,j,k,l = a.length;
        for(i = 0;i<l-1;i++)
        {
            for(j = 0;j<l-i-1;j++)
            {
                if(a[j]>a[j+1] && so == 1)
                {
                    k = a[j];
                    a[j] = a[j+1];
                    a[j+1] = k;
                }
                if(a[j]<a[j+1] && so == -1)
                {
                    k = a[j];
                    a[j] = a[j+1];
                    a[j+1] = k;
                }
            }
        }
        for(i = 0;i<l;i++)
            System.out.println (a[i]);
    }
    public static void main(int so)
    {
        int a[] = {2,8,11,9, 4,3,1,5,7,6, 10,14,17,18, 12,20,13, 15,16,19};
```

```

        bubbleSort(a,so);
    }
}

```

Q12. In Piglatin a word such as KING becomes INGKAY, TROUBLE becomes OUBLETRAY as so on. The first vowel of the original word becomes the starting of the translation and proceeding letter being shifted towards the end and followed by AY. Word that begins with a vowel is left unchanged. Write a class to accept a word and convert in to Piglatin word. Define a function int vowelPos(String) to find out the position of first vowel in the given word.

```

public class ques12
{
    public static int vowelPos(String word)
    {
        int i;
        for(i = 0;i<word.length();i++)
        {
            char c = word.charAt(i);
            if(c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
                break;
        }
        return(i);
    }
    public static void main(String word)
    {
        int p = vowelPos(word);
        if(p == 0)
            System.out.println (word);
        else
            System.out.println ("Piglatin word " + word.substring(p) + word.substring(0,p) + "AY");
    }
}

```

Q13. Write a MENU Driven program using separate function to calculate and return the answer of the following options from a given sentence:

a) Total number of digits present in it.

- b) Total number of small letters and capital letters present in it.
- c) Total number of alphabets used in it.
- d) Total number of vowels presents in it.
- e) Total Number words present in that sentence.

Q14. Write a class to display each word of the string in reverse order. e.g. If input is "India is my Country" output is "aidnI si ym yrtnouC". Use following function in your class:

public String reverse(String word) which will return the reverse word.

public void newstr(String Sentence) Receive a sentence and reverse the each word in the sentence.

```
public class ques14
{
    public static String reverse(String word)
    {
        String revWord = "";
        for(int i = word.length()-1;i >= 0;i--)
            revWord += word.charAt(i);
        return(revWord);
    }
    public static void newstr(String Sentence)
    {
        Sentence += " ";
        int p = 0;
        while(p < Sentence.length())
        {
            int x = Sentence.indexOf(' ', p); //findout the space position
            String w = Sentence.substring(p, x); //extracting word
            System.out.print (reverse(w) + " ");
            p = x + 1;
        }
    }
}
```

Q15. Design a program to overload a function rect()

void rect(int n,char ch) With one integer argument and one character type argument draw filled square of side n using the character stored in ch.

void rect(int l, int b, char ch)- With two integer argument and 1 character argument draw a filled rectangle of length l and breadth b using the character stored in ch. Get the following output.

@ @ @ @ @

@ @ @ @ @

A

B B

B B

C C C

C C C

C C C

D D D D

D D D D

D D D D

D D D D

public class ques15

{

private static void rect(int n,char ch)

{

for(int i = 1;i<= n;i++)

{

for(int j = 1;j<= n;j++)

{

System.out.print (ch);

}

System.out.println ();

}

}

public static void rect(int l,int b,char ch)

{

```

for(int i = 1;i<= l;i++)
{
    for(int j = 1;j<= b;j++)
    {
        System.out.print (ch);
    }
    System.out.println ();
}
}

public static void main()
{
    rect(2,5,'@');
    rect(1,'A');
    rect(2,'B');
    rect(3,'C');
    rect(4,'D');
}
}

```

Some Complex/Hard Questions

Q. Write a program to print all elements of Fibonacci series (between 1 to 1000),which is also a prime number using a defined function.

Q. N is a perfect number if the sum of all factors of the number (including 1 but excluding the number) is equal to N. For e.g. $6 = 1+2+3$. The class numberProblem below contains some methods that help you to work with perfect number. Study the class and then answer the following questions.

Public class numberProblem

```

{
    int sumOfFactors(int N)
    {
        // which return sum of all the factors of the number N.
    }

    boolean isPerfect(int N)

```

```

{
    // return true if the number N is perfect and false otherwise.
}

void nosBelow(int lim)
{
    // which prints out all perfect numbers less than lim.
}
}

```

Define the class numberProblem and the definitions of all the methods given in that class.

```

public class numberProblem
{
    int sumOfFactors(int N)
    {
        // which return sum of all the factors
        // of the number N.
        int s = 0;
        for(int i = 1; i <= N/2; i++)
            if(N%i == 0)
                s += i;
        return(s);
    }

    boolean isPerfect(int N)
    {
        // return true if the number N
        // is perfect and false otherwise.
        int sum = sumOfFactors(N);
        if(sum == N)
            return(true);
        else
            return(false);
    }
}

```



```

void nosBelow(int lim)
{
    // which prints out all perfect
    // numbers less then lim.
    for(int i = 1;i<= lim;i++)
        if(isPerfect(i))
            System.out.println (i);
    }
}

```

Q. A Manager of an Internet Service Provider company wants to analyze the system usage from the log records to find the utilization of the system. He wants to know:

For how long did each user use the system?

When the user wants to use the system he must login to the system (starttime e.g. 12:30) and after finishing the work he must log out off the system (endtime e.g. 18:10).

The time format is the 24 hour system hour and minute.

You may also assume that the data is valid and a user will log on less then 24 hours.

If the endtime (e.g. 2:30) is less then starttime (e.g. 19:20) it means that the user logged in over the night.

A class called Time has been defined to calculate the time related functions. Some of the function/months in Times are shown below:

```

class times
{
    int hh,mm;
    public times()
    { // default constructor to initialize value.
    }
}

```

```

public void readtime()
{ // to read time in 24 hour format as HH MM
}

public void disptimeansi()
{ // to display time in 24 hour format as HH MM
}

public int timetominutes(times)
{ // to find total number of minutes in HH MM
}

public void minutestotime(int)
{ // to convert total number of minutes into HH MM
}

public void diff(times endtime, times starttime)
{ // to find the diff. between endtime and starttime in Hrs. and Min
}
}

```

(a) Specify the class Times giving the details of the constructor, int timetominutes(times), void minutestotime(int), void diff(times endtime, times starttime) only. You may assume that the other functions are written for you. You do not need to write the main function.

(b) What care do you need to take while designing void diff(times endtime, times starttime)?

1. Write a program to accept a sentence and reverse each word in that sentence and print it. Program should also count number of palindrome word present in that sentence.

Define a class that has the following functions.

String reverseWord(String word) -> which reverse the word and return it.

boolean isPalindrom(String word) -> which check and return true if the word is palindrome else return false.

public void main(String Sentence) -> which print the sentence after reversing each word, and then print number of palindrome word present in it.

class palindromeCount

```

{
String reverseWord(String word)
{
String reverseWord = "";
int l = word.length();
for(int i = l-1;i >= 0;i--)
reverseWord = reverseWord + word.charAt(i);
return(reverseWord);
}
boolean isPalindrom(String word)
{
String w = reverseWord(word);
return(word.equals(w));
}
public void main(String Sentence)
{
Sentence += " ";
int p = 0, ctr = 0;
while(p < Sentence.length())
{
int x = Sentence.indexOf(' ', p);
String word = Sentence.substring(p, x);
System.out.print (reverseWord(word) + " ");
if(isPalindrom(word) == true)
ctr++;
p = x+1;
}
System.out.println ("\nNo of Palindrom Word : "+ctr);
}
}

```

2. A number is called Armstrong number if the sum of cube of each digit of the number is equal to that number. (e.g. 153 is a Armstrong number because $153 = 1^3 + 5^3 + 3^3$).

Define a class armStrong that has the following functions.

private int sumOfDigit(int N) -> which return the sum of cube of each digits present in N.

public static void main(int x, int y) -> which display all the arm strong number between the range x and y.

```
class armStrong
{
    private static int sumOfDigit(int N)
    {
        int sum = 0;
        while(N>0)
        {
            int digit = N%10;
            sum = sum + (digit*digit*digit);
            N = N/10;
        }
        return(sum);
    }
    public static void main(int x, int y)
    {
        for(int number = x; number<= y; number++)
        {
            if(number == sumOfDigit(number))
                System.out.println ("Armstrong Number is " + number);
        }
    }
}
```

3. Write a program to input a positive natural number N and output all combination of consecutive natural numbers which added up to give N

Example: If the N is 15 then output should be

1 2 3 4 5 = 15

4 5 6 = 15

7 8 = 15

Define a class consecutive that has the following functions.

private static void printSet(int i, int j) -> which print one line series of numbers starting from i to j, as mentioned in the above format.

public static void main(int N) -> which will print all number of possibilities upto N, as mentioned in the above example.

public class consecutive

{

private static void printSet(int i, int j)

{

int s = 0;

for(int k = i;k<j;k++)

{

s+= k;

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

}

System.out.println (" = "+s);

}

public static void main(int N)

{

int i,j,s;

for(i = 1;i<N;i++)

{

s = 0; j = i;

while(s<N)

{

s = s+j; j++;

}

if(s == N)

{

printSet(i,j);

```

    }
}
}
}

```

1. Write a program to accept a sentence then convert it to Title Case. That means first character of each word should be capital and remaining are small letter. (e.g. MY NAME IS AMITABH become My Name Is Amitabh)

Define a class that has the following functions.

private String properWord(String word) -> which convert the word to proper case (AMIT becomes Amit) and return it.

public void main(String Sentence) -> which print the given sentence to in the above mention format.

```

public class properSentence
{
    private String properWord(String word)
    {
        word = word.toLowerCase();
        char firstChar = Character.toUpperCase (word.charAt(0));
        String newWord = firstChar + word.substring(1);
        return(newWord);
    }
    public void main(String Sentence)
    {
        Sentence+= " ";
        int p = 0;
        while(p<Sentence.length())
        {
            int x = Sentence.indexOf(' ',p);
            String word = Sentence.substring(p,x);
            System.out.print (properWord(word)+" ");
            p = x+1;
        }
    }
}

```

```
}
```

```
}
```

2. Write a program to determine if the input number is a lucky number or not. A lucky number is a number whose digits when added (till a single digit is obtained) it equals to 1. For example,

$4567 = 7+6+5+4 = 22$ $22 = 2+2 = 4$ 4 is not $= 1$

So 4567 is not a lucky number.

$4627 = 7+6+2+4 = 19$ $19 = 1+9 = 10$ $10 = 1+0 = 1$

So 4627 is a lucky number.

Define a class luckyNumber that has the following functions.

private int sumOfDigit(int N) -> which return the sum of digits present in N.

public void isLucky(int number) -> which check and display a proper message, if the given number is Lucky number or not.

class luckyNumber

```
{
```

```
public int sumOfDigit(int N)
```

```
{
```

```
int sum = 0;
```

```
while(N>0)
```

```
{
```

```
int digit = N%10;
```

```
sum = sum+digit;
```

```
N = N/10;
```

```
}
```

```
return(sum);
```

```
}
```

```
public void isLucky(int number)
```

```
{
```

```
int digit;
```

```
while(number>9)
```

```
{
```

```
number = sumOfDigit(number);
```

```
}
```

```

if(number == 1)
    System.out.println ("Lucky Number");
else
    System.out.println ("Not Lucky Number");
}
}

```

3. A class clock has following members:

Instance variable: hour and minute of integer type.

Member function/methods:

public clock(int,int) -> constructor to initialize hour and min.

public void showTime() -> to display the time.

public void addTime(clock,clock) -> to add to clock objects.

Write another class time with main() function which create two different objects FT, ST print their sum using the above clock class function. If FT = 6 Hrs 35 Min and ST = 3 Hrs 45 Min then output 10 Hrs 20 Min.

```

class clock
{
    int hour, min;
    public clock(int h,int m)
    {
        hour = h;
        min = m;
    }
    public void showTime()
    {
        System.out.println (hour + " Hrs " + min + " Min");
    }
    public void addTime(clock T1, clock T2)
    {
        min = (T1.min+T2.min)%60;
        hour = (T1.hour+T2.hour) + (T1.min+T2.min)/60;
    }
}

```



```

}

public class time
{
    public static void main()
    {
        clock FT = new clock(6,35);
        clock ST = new clock(3,45);
        clock output = new clock(0,0);
        output.addTime(FT,ST);
        output.showTime();
    }
}

```

1. Convert each word to Piglatin word in a sentence.

Define a class that has the following functions.

private static int vowelPos(String word) -> which find and return the first vowel position in the word.

private static String Piglatin(String word) -> which convert the word to Piglatin word and return it.

public void main(String Sentence) -> which print the sentence after converting each word to Piglatin word.

```

public class pigSentence
{
    private static int vowelPos(String word)
    {
        for(int i = 0;i<word.length();i++)
        {
            char c = word.charAt(i);
            if(c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
                break;
        }
        return(i);
    }

    private static String Piglatin(String word)

```

```

{
    int p = vowelPos(word);
    String newWord = "";
    if(p == 0)
        newWord = word;
    else
        newWord = word.substring(p) + word.substring(0,p)+"A";
    return(newWord);
}

public static void main(String Sentence)
{
    Sentence+ = " ";
    int p = 0;
    while(p<Sentence.length())
    {
        int x = Sentence.indexOf(' ',p);
        String word = Sentence.substring(p,x);
        System.out.print (Piglatin(word) + " ");
        p = x+1;
    }
}
}

```

2. N is a perfect number if the sum of all factors of the number (including 1 but excluding the number) is equal to N (for e.g. $6 = 1+2+3$). N is a prime number if it is divisible by 1 and itself.

Define a class numProblem that has the following functions.

int nthPrime(int N) -> which return the nth prime number(first prime number is 2) (e.g. 3rd prime number is 5)

void perfectNosBelow(int N) -> which first print out the nth prime number and then print out all perfect numbers less than the nth prime number.

```

public class numProblem
{
    public static int nthPrime(int N)

```

```

{
    int ctr = 0, Prime = 2;
    while(true)
    {
        boolean flag = true;
        for(int i = 2; i < Prime; i++)
        {
            if(Prime % i == 0)
            {
                flag = false;
                break;
            }
        }
        if(flag == true) ctr++;
        if(ctr == N) break;
        Prime++;
    }
    return(Prime);
}

public static void perfectNosBelow(int N)
{
    int X = nthPrime(N);
    System.out.println(N + "th Prime no. : " + X);
    for(int i = 1; i < X; i++)
    {
        int sum = 0;
        for(int j = 1; j < i; j++)
        {
            if(i % j == 0)
            {
                sum += j;
            }
        }
    }
}

```

```

    }
}
if(sum == i)
{
    System.out.println(i);
}
}
}
}
}
}

```

3. Your computer science teacher is trying to analyze the performance of the class in the previous exam. He has a class called performance that contains the marks of 50 students. These are not sorted. He wants to find two quantities. (i) Mode - The most frequently occurring mark in the class. If two or more marks occur equally frequently then the highest of these marks is the mode. (ii) Mode frequently - Frequency at mode. Important: You are not allowed to sort the marks.

Some of the instance variable and member functions of the performance class are given below.

mark[] -> an array to store Marks of 50 students

mode -> to store the Mode

freqatmode -> the frequency at mode.

void readmarks() -> for reading the marks into the array.

void calcmodeandFrequency() -> a single function that calculates both mode and frequency at mode.

import java.io.*;

class performance

```

{
    int mark[] = new int[10];
    int mode = 0;
    int freqatmode = 0;
    public void readmark() throws IOException
    {
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader in = new BufferedReader(isr);
        for(int i = 0; i < 10; i++)
        {

```

```

        System.out.print ("Enter Marks : ");
        mark[i] = Integer.parseInt ( in.readLine() );
    }
}
public void calcmodeandfrequency()
{
    int f,i,j;
    for(i = 0;i<10;i++)
    {
        f = 0;
        for(j = 0;j<10;j++)
        {
            if(mark[i] == mark[j])
                f++;
        }
        if(f>= freqatmode && mark[i]>= mode)
        {
            mode = mark[i];
            freqatmode = f;
        }
    }
}
public void main() throws IOException
{
    readmark();
    calcmodeandfrequency();
    System.out.println ("Marks : "+mode);
    System.out.println ("Frequency : "+freqatmode);
}
}

```