1) write a Java program to find hundred (100) factorial and roots of quadratic equation?

A) CODE:-

```
Class factorial {

    static void fact (int n)
    {
        int res[] = new int[500];
        res[0] = 1;
        int res-size = 1;
        for (int x=2; x ≤=n ; x++){
            res_size = multiply (r, res, res_size)
            system.out.println ("factorial of given number
                                    is"); }
            for (int i= res_size-1; i>=0; i--)
                system.out.println(res[i]);

        }

    static int multiply ( int x , int res[ ], int res size)
    {
        int carry= 0;
        for (int i=0; i<res_size; i++)
        {
            int prod = res[i]* x + carry;
            res[i ]= prod %10;
            carry = prod / 10;
        }
```

```
while (carry! =0)
  {
     res[res_size] = carry%10;
     carry= carry/10;
     res_size++
   }
   return res_size;
   }
 }
class QuadraticRoot{
     double a,b,c;
     Quadratic Root[double x, double y, double z){
          a=x;
          b=y;
          c=z;
     }
   void Root() {
     double root1, root2;
     double deteeminant = b*b-4*a*c;
     if (deteeminant >0){
   root1 = (-b+math.sqrt(deteeminant))/(2*a);
   root2 = (-b+math.sqrt(deteeminant))/(2*a);
   system.out.println("\n root1=%.2f and root2=%.2f",
                                        root1, root2)
   }
  else f (deteminant==0){
    root1 = root2 = -b/(2*a);
    system.out.println("\n root1=root2 =%.2f,",root1);
  }
```

```
class Main
{
  Public static void main (String args[])
  {
    factorial f1 = new factorial ( );
    f 1. fact (100);
    Quadratic Root Q1 = new QuadraticRoot (2,3,4,5,6)
     Q1. root();
  }
```

output

```
Factorial of given number is
933226215443944152681699238852667004907
- - - - - . . . . 00000000000000000000 000.
root 1 = -0.87+1.30;
root2 = -0.87 -1.30;
```

2) write a Java program that uses both recursive and non recursive functions to print the nth value in the fibannoci series/sequence where n=100, then what is Your observation in recursive fibanocci series.

CODE

A

```
import java.io.DataInputStream;
class fibo
{
```

```
Public void fibo-non-recursive (Long n)
{
   Long f1=0, f2=1, f3=0, i=1;
   System.out.printh(f1);
   System.out.printh(f2);
   while (i<=n)
    {
      f3=f1+f2;
      system.out.printhn(f3];
      f1=f2;
      f2=f3;
      i++;
    }
   System.out.printh("sum of fibonocci series upto" +n" is"+f3);
}

Public long fibo_recursive (long n)
  {  if (n<=2)
          system.out.printhn(n);
      else
          System.out.printh(fibo-recursive(n-1)+ fibo_recursive(n-2));
      }
   }
```

```
Class fibomain
{
   Public static void main (String args[])
   {
      long n= 100;
      fib obj = new fibo ();
      DataInput Stream x= new DataInputStream(system.in);
     try {
     system.out.println("enter number to print fibonacci series
                                  upto N:");
     n = long.parselong(x.readLine());
     System.out.println("non recursive press Enter to start");
     x = readLine();
     obj.fibo_non_recursive(n);
     System.out.println("\n Press Enter to continue...");
    x.readLine();
    system.out.println("calling Recursive return_type:");
    System.out.println(obj.fibo_recursive(n));
   system.out.println("\n calling Recursive void-type:");
   obj.fibo_recursive2(n);
 }
catch (exception e) {

     }
```

OUTPUT OBSERVATION

A non recursive function will generate 100 number fibanocei series very faster (may be within a second) then, the recursive function

* A recursive function fibanocci series up to 100 will take around 15-45 minutes, based on the computer calculation speed. Because recursion takes a lot of time in execution of series of instructions for tasks such as calling a function again and again drecursively Jumping to the called version functions, requires a lot of CPU line and memory space

3) Write a Java program that prompts the user for an integer and then prints out sum of all prime numbers up to the integer where n=2 million

CODE

```
import java.io.DataInputStream;
class prime
{
      Private   int sum=0, i=1;
      Public int prime(long n)
      {
            while (i<=n)
         {
               int J=1;
               int f=0;
               int c=2;
```

```
while (k<j)
  {   if (j%k==0)
      {   f=1;
          break;
      }
      k++
   }
   if (f==0)
    {
     system.out.println(j + " is a prime");
      sum=sum+j;

    }
    i++
  }
   return (sum);
   }
 }
class primemain
 {   Public static void main (String args[])
    {
      long n=100;
     prime obj = new prime();
     DataInputStream x = new DataInput Stream(system.in);
     try {
      system.out.print("enter number to print sum of
                        primes of N:");
      n= long.passelong(x.readLine());
     System.out.println("The sum of primes upto "+n+"is:"+
                                              Obj.print(n))
```

```
}
catch (exception e) { }
}
```

output

```
19999737 is a prime
1999771 is a prime
19997799 is a prime
1999 817 is a prime
1999 819 is a prime
1999 853 is a prime
1999 859 is a prime
1999 867 is a prime
1999 871 is a prime
1999 889 is a prime
1999 891 is a prime
1999 957 is a prime
1999 969 is a prime
1999 979 is a prime
1999 993 is a prime
The sum of primes upto 2000000 is : 1179908155 ] ×
```

4) write a Java program that reads a line of integer, and then display each integer and the sum of all Integers (use String Tokenizer of Java. util)

CODE

```
import Java.util.*;
import Java.io.DataInputStream
class StrTokenmain
{
   Public Static void main (string args[])
   {
     DataInput stream x = new DataInput Stream(system.in);
     int sum=0, n;
     string s=" ";
     try
      {
       system.out.println ("enter numbers seperated by
                                         space:");
            s= x.readLines();
      }
     catch (exception e) {}
     string Tokenizer str = new stringTokenizer(s);
     while (str.has MoreTokens ())
     {
       n= integer.parseInt ( str.nextToken());
       system.out.println(n);
       sum =sum+n;
     }
     System.out.println ("The sum of the integers is: "+sum );
   }
}
```

OUTPUT

```
enter numbers seperated by space: 10 20 30 40 50 60 70
                                  80 90 100.

10
20
30
40
50
60
70
80
90
100
Sumof integers is 550
```

5) Prime nos sum

```
import java.util.*;
 import java.util.scanner;
class Sumof primeNumbers {
     void sum(int n) {
        System.out.println("1");
        int i,j,m=0;
        long sum=2;
        for(j=3; j<i; j+=2){
           if (i%j==0){
              m++;}
           if (m==1) {
               break;
          }
      }
if (m==0) {
   sum +=i;
   system.out.printh (i+ "is a prime no");}
    }
   System.out-printhu (sum + "=sum ");
    }
  }
Public class Prime Numbers {
  Public static void main (string[] args)
  {
      Scanner p =new Scanner (system.in);
      system.out.println( "enter n");
      int n = p.nextInt();
      sum of PrimeNumbers m=new sumof PrimeNumbers();
      m. sum (n);
    }
  }
```

output

```
19998859 is prime
1999867 is prime
1999871 is prime
1999889 is prime
1999891  is prime
1999957 is prime
1999969  is prime
1999979 is prime
1999993 is prime

142913828922 isSUM
```

5) write a Java program to make frequency count of words in a given Test?

(A)

```
import java.io*;
class freq
{
    Public static void main(String args[])throw Exception
  {
    Input Stream Reader r=new InputStreamReader(system.in);
    Buffered Reader br=new BufferedReader(r);
    String str;
    systemout.println("entera line of txt with repeated
                                  words:\n");
   str = br.readLine();
   String strarr[]=str.split(" ");
   System.out.println("\n the frequency of words in the
                                     test:\n");
  for (String i : strarr)
    { int c=0; j;
      for (j=0; j<strarr.length; j++)
       {
         if (iequalsIgnorecase(strarr[j]))
         {
           c++;
           strarr[j]="0";
         }
       }
      if(!i=equals("0"))
       System.out.println(i+" "+c); } }
```

5) write a Java program that displays the number of characters lines and words in text files.

CODE

```
import java.io.* ;
class file2
{
  file writer fw=null;
  BufferedWriter bw=null;
  Printwriter pw=null;
  String s;
  Input Stream Reader ir = new InputStreamReader (system.in);
  BufferedReader br=new BufferedReader(ir);
  Public void createfile( ) throws io Exception
  {
      try
      {
       fw =new fileWrite ("micosmart.txt", true)
       bw = new BufferedWriter (fw);
       pw = new PrintWriter (bw);
      System.out.println ("enter a multitext & type STOP to
                  save & exit from the file : \n");
      while(true) {
         s=br.readLine ()
         if (s.equalsIgnorecase ("stop"))
            break;
         pw.println(s);
      }
System . out . println ("Data successfully appended into file");
pw. flush ( ) ;
}
finally
  {
     try
     {
        pw.close();
        bw. close();
        fw. close();
      }
     catch (IOException io) { }
     }
   }
  }
  Public void readfile () throws IOException
  {
       int ch;
     fileReader fr = null;
     try
     { fr = new FileReader ("microsmart".txt ");
     }
    catch (fileNotFoundException fe) {
         System.out.println ("file not found");
         }
        system.out.println ("\n");
        while ((ch = fr.read ()!= -1)
         System.out.print((char)ch);
         fr.close ();
     }
```

```
Public void count() throws IOException
{
   int ch, w=0, l=0, c=0
   file Reader fr=null;
   try
   {
     fr =new fileReader ('microsmall.txt");
   }
   catch ( FileNotFound Exception fe)
   {
        System.out.println ("file not found");
   }
   while (Ch = fr.read ()!. = -1)
   {
       if ( Ch== ' ')
         w++;
       else if (ch=='\n')
          {
               i++;
               w++;
          }
          else
             c++;
       }
       System out.println ("\ncharacters: "+ ((-i)+"\nwords:"
                                          +w+"\nLines :"+i);

       fr.close ();
   }
}
```

```
class file2maim
{
  Public static void main (string args[ ]) throws IOexception{

        file2 n= newfile2();
            n.createfile();
            n.readfile();
            n.count();
        }
    }
```

OUTPUT

```
enter a multiline Text & type stop to save & exit from file.
microsmart computers from
 tender is one and only
 computer centre, which is
 a trustworthy for the all
 the academic students of
 BSC computer B.com computer
 engineering CSC, IT, DS departments
  STOP
 Data successfuly appendend into file
 microsmart computers from
 tender is only one
 computer center, which is
 a trustworthy for all
 the academic students of
 BSC computer, B com computer
 engineering CSC IT, DS depalments
```

characteers: 164

words : 28

lines : 7

(8) write a programm that creates 3 threads first thread display "good morning" every one second, thread displays "Hello" for every two seconds and third thread displays "welcome" for every 3 seconds?

CODE

(A)
```
class Thread extends Thread
 {
     string s;
      int n;
     thread (string ss, int i)
       {
           s = ss;
           n = i;
        }
       Public void run()
       {
           for(int i=1 ; i<=10 ; i++)
           {
               try
               { System.out.println(s);
                  Thread.sleep(n);
               }
               catch(exceptione) {}
              }
           }
        }
```

```
Class threadmain
{
   Public static void main (string args[])
   {
    thread t1 = new thread ("T1 goodmorning",1000);
    thread t2 = new thread ("T2 Hello" ,2000);
    thread t3 = new thread ("T3 welcome", 3000);

    t1.start();
    t2.start();
    t3.start();
   }
}
```

OUTPUT

```
T1 good Morning
T3  welcome
T2  Hello
T1  good Morning
T3  welcome
T1  goodmorning
T2   Hello
T1  goodmorning
T1  goodmorning
T3    welcome
T2    Hello
T1    goodmorning

T1   good morning
T2     Hello
T1  good morning
T2    Hello
T1    good morning
T3    welcome
T1    good morning
T2    Hello
T3       welcome
T2       Hello
T2     Hello
T3    welcome
```

9) Write a Java program that correctly impuments produces consumer problem using the concept of inter thread communication?

(A) CODE

```
class Thread1
{
    int n;
    boolean valueset=false;
    synchronized int get()
     {
          if(!valueset)
           try
           { wait();
          }
         catch(exception e)
          {
             Systemout.println("exception occurs at :"+e);
          }
         System.out.println("get"+n);
         try
          {  thread.sleep(1000);
          }
          catch(Exception e)
           {
            System.out.println("exception occurs at !"+e);
           }
          valueset=false;
          notify();
          return n;  }
```

```
Synchronized int put(int n)
{
    if(valueset)
        try
        {
         wait();
        }
        cath (exception e)
        {
         System.out.printh("exception occues at: "+e);
        }
      this.n=n;
      valueset=true;
    system.out.println("put"+n);
   try
   {
      Thread.sleep(1000);
   }
   catch (exception e)
    {
     system.out.println("exaption occurat: "+e);
    }
   notify();
   return n;
  }
 }
class producer implements Runnable
{ Thread1 t;
   producer(Threaad1 t)
   { this.t=t;
       new thread(this, "producer"),.start();
   }
```

```
Public void run()
  {
      int i=0;
      while (true)
        {
          t.put(i++);
        }
     }
  }

class Consumer implements Runnable
{
    Thread1 t;
   consumer (Thread1 t)
     {
       this.t=t;
       new thread (this, "consumer").start();
     }
     Public void run()
       {    int i=0;
            while (true)
             {     t.get();
             }
           }
         }
    class ProducerConsumer
  {
    Public static void main(String[] args)
     {
         Thread1 t = new Thread1();
          new producer(t);
          new consumer(t);
          System.out.println("press control+c to exit");}}
```

OUTPUT

```
Press ctrl+c to exit
put 0
get 0
Put 1
get 1
Put 2
get 2
Put 3
get 3
Put 4
get 4
Put 5
get 5
```

(10) write a Java program to print pascal triangle in optimal way?

(A) CODE

```
Public class PascalTriangle {
        static int factorial (int n ) {
              int f ;
              for( f=1 ; n>1 ; n--){
                     f =n ;
              }
              return f ;
        }
        static int ncr(int n, int r){
              return factorial(n)/ factorial(n-r) * factorial(r));
        }
        Public static void main (String args[ ]){
              System.out.println( );
              int n,i,j;
n = 5;
for (i=0; i<n; i++) {
    for (J=0; j<n-i; j++) {
        System.out.print(" ");
    }
    for (J=0 ; j<=i ; j++) {
        system.out.print(" "+ ncr(i, j));
    }
    system.out.println();
} }
```

OUTPUT

```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5  10  10   5   1
```

7) write a Java code to convert infix to postfix expression and evaluate it?

A) CODE

```
import java.util.stack;
class Test
{ static int Prec(char ch)
  { switch (ch)
    { case '+';
      case '-';
        retun 1;
```

```
case '*':
 case '\':
      return 2;

 case   '^':
         return 3;

   }
   return -1;
  }
 string  infixToPostfix(string exp)
  {
  string result = new string ("")
  Stack <character> stack  =new stack <>();
  for (int i=0; i< exp.length(), ++i)
   {
     char c = exp.charAt(i);
     if (character.isletterordigit(c))
         result += c;
       else  if  ( c == '(')
            Stack.push(c);
       else if  ( c == ')')
       {
        while (!stack is empty() && stack.peek()!
                                              = '(')
              result += stack.pop();
              stack.pop();
       }
     else {
        while(!stock.isempty() && prec(c)
                              <= prec(stack.peek())){
```

```
    result + = stack.pop( );
   }
   Stack.push (c);
  }
 }
while (!stack is Empty()){
      if (stack.peek() = = '(')
          return = 'Invalid Expression'';
          result + = stack.pop();
       }
     return result;
   } }
 class Test2
 {
  static int evaluate Postfix (string exp)
   {
    stack <Integer> stack = new  stack<>();
    for (int i=0; i<exp.length(); i++)
    {
     char c = exp.charAt(i);
     if (character.isDigit(c))
      stack.push (c);
     else {
      int val1 = stack.pop();
      int val2 = stack.pop();
      switch (c)
      {  case '+':
         Stack.push (val2+val1);
         break;
case '-':
 stack.push(val2-val1);
 break;
case '/':
 stack.push(val2/val1);
  break;
case '*':
 stack.push(val2* val4);
 break;
}}}
return stack.pop();
}
}
class Test
{ public static void main (string[] args)
 {   Test t2 = new Test2();
     Test t1 = new Test1();
     string exp1 = "(2+(3* 1))-9";
     system.out.println("Infixto Postfix:"+t1.infixTopostfix
                                               (exp1));
     string exp2 = "231* +9- ";
     system.out.println("postfix evaluation:"+t2.evaluate
                                               Postfix(exp2));
 }}
```

OUTPUT

```
infix To Postfix: 231* +9-
Postfixevaluation: -4.
```