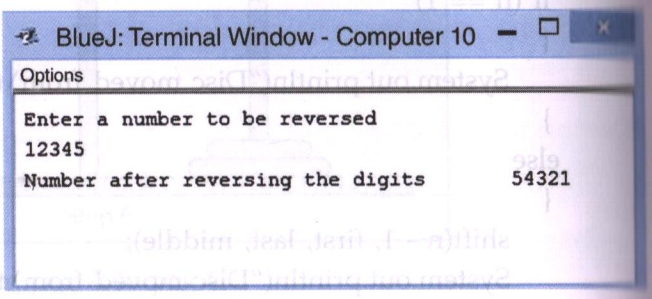
Solved Exercises

Prog. 1:

*Write a program to input a number and reverse the digits of the number by using **recursive** function. Display the new number.*

```
// A sample program to reverse a number by using Recursive function
import java.util.*;
class Reverse
{
public static int rev(int n,int r)
{
if(n==0)
    return(r);
    else
    {
    r=r*10+n%10;
    return(rev(n/10,r));
     }
}
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
System.out.println("Enter a number to be reversed");
int num=in.nextInt();
int res=rev(num,0);
System.out.println("Number after reversing the digits\t"+res);
}
}
```

Output:

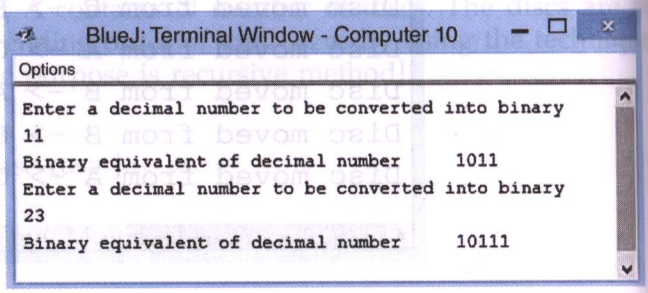


Prog. 2:

*Write a program to input a decimal number (base 10) and convert it into its binary equivalent by using the **recursive** function. Display the binary number.*

```
// A program to convert a decimal number into its binary equivalent by using Recursive
   function
import java.util.*;
class Binary
{
public static int recbin(int n,String s)
{
if(n==0)
    return(Integer.parseInt(s));
    else
        {
            s=Integer.toString(n%2)+s;
            return(recbin(n/2,s));
        }
```

Output:



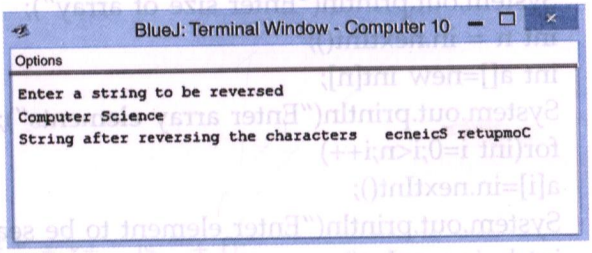
```
}
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
System.out.println("Enter a decimal number to be converted into binary");
int num=in.nextInt();
int res=recbin(num,"");
System.out.println("Binary equivalent of decimal number\t"+res);
}
}
```

Prog. 3:

Write a program to input a String and reverse it by using recursive function. Display the new String formed after reversal.

```
// A sample program to reverse a String by using Recursive function
import java.util.*;
class Str_rev
{
public static String reverse(String s, int p, String rev)
{
if(p<0)
    return(rev);
    else
    {
        rev=rev+s.charAt(p);
        return(reverse(s,p-1,rev));
    }
}
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
System.out.println("Enter a string to be reversed");
String st=in.nextLine();
String res=reverse(st,st.length()-1,"");
System.out.println("String after reversing the characters\t"+res);
}
}
```

Output:



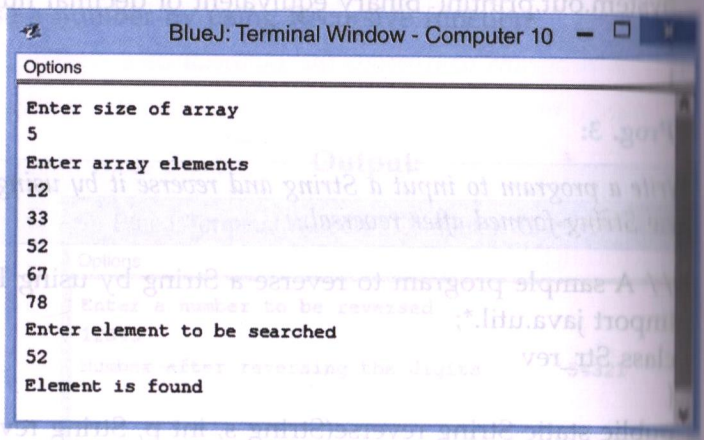
Prog. 4:

Write a program to input a set of 'n' numbers (ascending order) in a Single Dimension Array. Enter a number and search whether the number is present among the set of numbers or not by using Binary search technique. If the number is present then display a message "Element is found" otherwise, display "Element is not found". Perform the task by using ***Recursive*** *technique.*

```
// A sample program to perform Binary search by using Recursive function
import java.util.*;
class B_Search
{
```

```
public static int bsearch(int x[],int l,int h,int ele,int f)
{
//x[] Array containing elements
//ele is element to be searched
//l and h are lower and upper boundary correspondingly
//f is the flag to set 1 if number is found
int mid;
mid=(l+h)/2;
if(l>h||f==1)
    return(f);
    else
    {
    if(ele==x[mid])
        f=1;
    if(ele<x[mid])
        h=mid-1;
    if(ele>x[mid])
        l=mid+1;
    return(bsearch(x,l,h,ele,f));
    }
}
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
System.out.println("Enter size of array");
int n = in.nextInt();
int a[]=new int[n];
System.out.println("Enter array elements");
for(int i=0;i<n;i++)
a[i]=in.nextInt();
System.out.println("Enter element to be searched");
int k=in.nextInt();
int l=0,h=n-1;
int f=0;
int res=bsearch(a,l,h,k,f);
if(res==1)
    System.out.println("Element is found");
    else
    System.out.println("Element is not found");
}
}
```

Output:



Prog. 5:

The factorial of a number 'n' is calculated as:
n! = n(n-1)*(n-2)*(n-3)*----------*1.*
A class Factorial is declared with the following details:
Class name *: Factorial*
Data members/Instance variables:
n *: to store a number (integer type) greater than zero*

f : *to store the factorial of a number entered by the user*

Member functions/methods:

Factorial( ) : *a constructor to assign 0 to n*

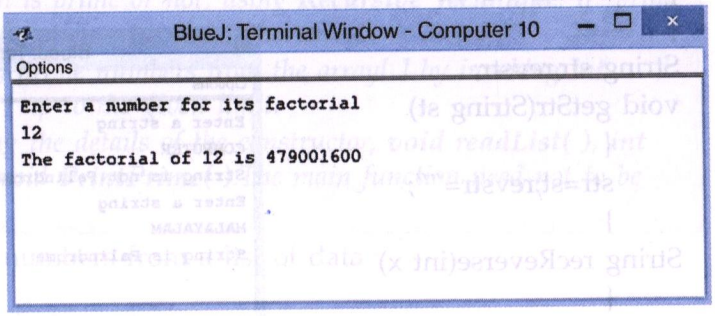
int fact(int num) : *to find and return the factorial of num (num>0) using* ***recursive technique*** *otherwise, it should return 1 if num=0*

void get(int x) : *to assign x to n and by invoking function fact( ) store the factorial of n into f. Display the result with a suitable message.*

Write the main( ) method to create an object to class Factorial and call the function get(int).

```
// A program to display the factorial of a number
import java.util.*;
public class Factorial
{
int n,f;
Factorial()
    {
    int n=0;
    f=0;
    }
int fact(int num)
    {
    n=num;
    if(n==0)
        return(1);
    else
        return(n*fact(n-1));
    }
void get(int x)
    {
    n=x;
    f=fact(n);
    System.out.println("The factorial of " + x+ " is " + f);
    }
public static void main(String args[])
{
int m;
Scanner in = new Scanner(System.in);
System.out.println("Enter a number for its factorial");
m= in.nextInt();
Factorial ob=new Factorial();
ob.get(m);
}
}
```

Output:



Prog. 6:

A class Revstr defines a recursive function to reverse a string and check whether it is Palindrome or not. The details of the class are given below:

Class name : *Revstr*

Data members/instance variables:

str : *to store string*

Revst : *stores the reverse string*

Member functions/methods :

void getStr( ) : *to accept the string*

void recReverse(int) : *to reverse the string using* ***Recursive Technique***

void check ( ) : *to display the original string, its reverse and check whether the string is Palindrome or not.*

Specify the class ***Revstr*** *giving the details of the functions* ***void getStr( ), void recReverse( )*** *and* ***void check( )****. The main( ) function need not be written.*

```
// A program to display reverse String using recursive function
import java.util.*;
public class Revstr
{
String str,revstr;
void getStr(String st)
      {
      str=st;revstr="";
      }
String recReverse(int x)
      {
      if(x==str.length())
          return(revstr);
      else
          {
          revstr=revstr+str.charAt((str.length()-1)-x);
          return(recReverse(x+1));
          }
      }
void check()
      {
      if(str.equals(recReverse(0)))
          System.out.println("String is Palindrome");
      else
          System.out.println("String is not Palindrome");
      }
public static void main(String args[])
{
String st;
Scanner in=new Scanner(System.in);
Revstr ob = new Revstr();
System.out.println("Enter a string");
st=in.next();
ob.getStr(st);
ob.check();
}
}
```

Output:

```
BlueJ: Terminal Window - Computer 10
Options
Enter a string
COMPUTER
String is not Palindrome
Enter a string
MALAYALAM
String is Palindrome
```

Prog. 7:

A class Prime_Series defines a recursive function to find the prime numbers from a list of data. The details of the class are given below:

Class name : Prime_Series

Data members/instance variables:

limit : to store the limit of the integers

arr[ ] : to create an integer array (maximum size 150)

Member functions/methods:

Prime_Series( ) : constructor to assign 0 to limit and arr[ ]

void readList( ) : to accept the limit and input the integers in array arr[ ] up to the limit.

*int (IsPrime, int q) : to check n is prime or not, using **Recursive Technique**. If prime then the function returns 1, otherwise 0.*

void PrintPrime( ) : to display prime numbers from the array[ ] by invoking int IsPrime( ) up to the given limit.

*Specify the class **Prime_Series** giving the details of the constructor, **void readList( ), int IsPrime(int num, int j, int f)** and void **PrintPrime( )**.The main function need not to be written.*

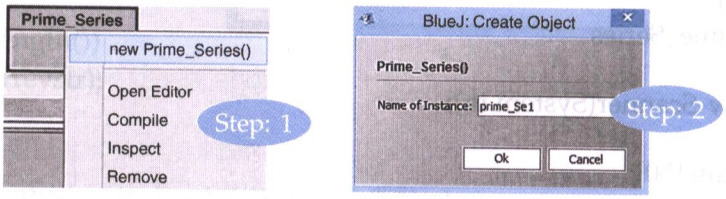
```
// A program to display prime numbers from a list of data
import java.util.*;
public class Prime_Series
{
Scanner in=new Scanner(System.in);
int limit;
int arr[]=new int[150];
Prime_Series()
    {
    int i;
    for(i=0;i<150;i++)
      {
      arr[i]=0;
      }
    }
void readList()
    {
    int j;
    System.out.println("Enter the number of elements to store:");
    limit= in.nextInt();
    System.out.println("Enter the elements:");
    for(j=0;j<limit;j++)
      {
      arr[j]=in.nextInt();
      }
    }
      int Isprime(int num,int j,int f)
      {
      if(j==num)
      return(f);
        else
          {
          if(num%j==0)
```

```
            f=0;
            return(Isprime(num,++j,f));
        }
    }
void PrintPrime()
    {
    int v;
    for(int i=0;i<limit;i++)
      {
      v=Isprime(arr[i],2,1);
      if(v==1)
        System.out.println("Number"+arr[i]+" is prime");
      }
    }
}
```

Execution of the program:

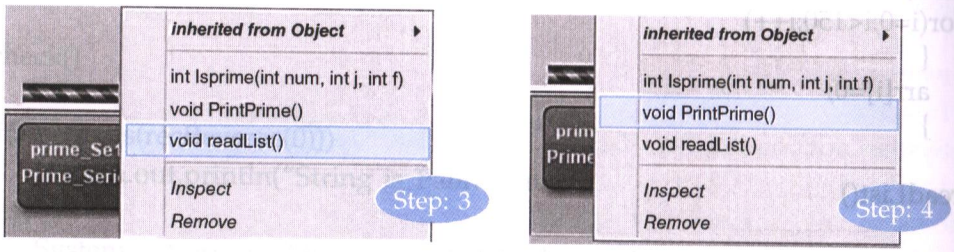
Step 1: Right click Prime_Series, and select newPrime_Series( )

Step 2: Click Ok on Create Object window.



Step 3: Select voidreadList( ) to enter the elements in the array

Step 4: Select and click PrintPrime( ).



Step 5: It displays the output on the Terminal Window.