



# Computer Project

Session - 2021 -22

---

M.C. KEJRIWAL VIDYAPEETH, LILUAH

*Subhranil Nandy | Roll- 3560*

*Class - XII | Sec - A*

---

## **ACKNOWLEDGEMENT**

I would like to express my gratitude to my teacher, Mr. Spondon Ganguly who gave me immense support during project development and provided me with the golden opportunity to do this wonderful project on Java programs, which helped me in doing a lot of brainstorming on coding and eventually built up my experience in programming and project work.

*Subhranil Nandy*  
*Subhranil Nandy*

## INDEX

Serial No.	Program	Page
1.	<b>Introduction to Object Oriented Programming</b>	4
2.	<b>Introduction to Java</b>	5
3.	<b>Hardware and software specifications</b>	6
4.	<b>Program to calculate frequency of ages of travellers</b>	7
5.	<b>Program to print histogram of frequency of characters in a sentence</b>	10
6.	<b>Program to change vowels to next alphabet and consonants to previous alphabet in a sentence</b>	16
7.	<b>Program to print all possible combinations of travelling to n cities starting from a particular city</b>	19
8.	<b>Program to print anagrams of a word</b>	26
9.	<b>Program to calculate Brun's Constant upto n terms</b>	29
10.	<b>Program to print all the perfect squares between 2 numbers</b>	32
11.	<b>Program to print the sum of a given series</b>	37
12.	<b>Program to print a 3*3 magic square and perform ascending row sort and descending column sort to the magic square</b>	40
13.	<b>Program to check the validity of a password based on the given conditions</b>	46
14.	<b>Program to check whether a string is balanced based on the given conditions</b>	50
15.	<b>Program to convert Roman numbers to decimal and vice versa</b>	53
16.	<b>Program to convert a decimal number to either binary, octal or hexadecimal based on user's choice</b>	58
17.	<b>Program to print the time in words</b>	63
18.	<b>Program to print the personal details of an employee with provident fund and gratuity</b>	67

<b>19.</b>	<b>Program to calculate total marks in exam</b>	<b>72</b>
<b>20.</b>	<b>Program to play ring game by piling ring and removing them</b>	<b>76</b>
<b>21.</b>	<b>Program to implement circular queue</b>	<b>81</b>
<b>22.</b>	<b>Program to convert Decimal number system to Binary/Octal/Hexadecimal equivalent and vice-versa</b>	<b>87</b>
<b>23.</b>	<b>Program to implement encryption and decryption of strings</b>	<b>94</b>
<b>24.</b>	<b>Bibliography</b>	<b>98</b>

## INTRODUCTION TO OBJECT ORIENTED PROGRAMMING

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which can contain data and code: data in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods).

A feature of objects is that an object's own procedures can access and often modify the data fields of itself (objects have a notion of "this" or "self"). In OOP, computer programs are designed by making them out of objects that interact with one another. OOP languages are diverse, but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

❖ Examples: Ruby, Scala, Smalltalk, Eiffel, Emerald, JADE, Self, Raku.

Languages designed mainly for OO programming, but with some procedural elements. Examples: Java, Python, C++, C#, Delphi/Object Pascal, VB.NET.

The general concepts of OOP are:

- Data Abstraction – Refers to the act of representing essential features without including the background details and explanations.
- Modularity – Property of a system that has been decomposed into a set of cohesive and loosely coupled modules.
- Polymorphism – Ability of a data or message to be processed in more than one form.
- Data Encapsulation – The wrapping up of data and functions (that operate on the data) into a single unit (called class).
- Inheritance – Capability of one class of things to inherit capabilities or properties from another class.

## INTRODUCTION TO JAVA

JAVA was developed by James Gosling at Sun Microsystems Inc in the year 1991, later acquired by Oracle Corporation. It is a simple programming language. Java makes writing, compiling, and debugging programming easy. It helps to create reusable code and modular programs. Java is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible. A general-purpose programming language made for developers to write once run anywhere that is compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can run on any Java Virtual Machine. The syntax of Java is similar to c/c++. Java is the name of an island in Indonesia where the first coffee (named java coffee) was produced. And this name was chosen by James Gosling while having coffee near his office.

Before learning Java, one must be familiar with these common terms of Java.

1. Java Virtual Machine (JVM): This is generally referred to as JVM. There are three execution phases of a program. They are written, compile and run the program.
2. Bytecode in the Development process: As discussed, the Javac compiler of JDK compiles the java source code into bytecode so that it can be executed by JVM. It is saved as .class file by the compiler.
3. Java Development Kit (JDK): While we were using the term JDK, when we learn about bytecode and JVM. So, as the name suggests, it is a complete Java development kit that includes everything including compiler, Java Runtime Environment (JRE), java debuggers, java docs, etc.
4. Java Runtime Environment (JRE): JDK includes JRE. JRE installation on our computers allows the java program to run, however, we cannot compile it.
5. Garbage Collector: In Java, programmers can't delete the objects. To delete or recollect that memory JVM has a program called Garbage Collector. Garbage Collectors can recollect the of objects that are not referenced.
6. ClassPath: The classpath is the file path where the java runtime and Java compiler look for .class files to load.

## **HARDWARE AND SOFTWARE SPECIFICATIONS**

- RAM – 4 GB
- CPU – 64-bit OS Intel(R) Core (TM) i3 5005U CPU @ 2.00 GHz
- OS – Windows 10 Pro
- JAVA SOFTWARE – BlueJ
- BLUEJ SOFTWARE – BlueJ Version 4.2.2

### **Question 1.**

Avoltri Travels were taking a group of 200 tourists to Balaji Temple.

A class clTravel is designed, some of whose functions/methods are shown below-

Class name clTravel

Data members/ instance variables: int arAge[200] – an array to store the age of 200 tourists.

int arFreqDist[5] – an array to store the number of people in various age groups: -

cell 0 – number of people upto 20 years

cell 1 – number of people between 21 to 40 years

cell 2 – number of people between 41 to 60 years

cell 3 – number of people between 61 to 80 years

cell 4 – number of people above 80 years

Member functions/ methods:

clTravel( ) Constructor

void fnReadAge( ) To input age of 200 tourists in arAge[]

void fnFrequency( ) To fill arFreqDist[ ] from arAge[ ]

void fnShowFreq( ) To show the frequency of age group in a table format

Specify the class clTravel giving details of the mentioned functions. You need to write the main()

method and create an object of the class and call the functions accordingly.

### **SOLUTION:**

### **ALGORITHM:**

1. START
2. Reads the ages of 10 travellers.
3. Calculates frequency of a particular age group.
4. Displays the age group with its frequency in a tabular format.
5. STOP

## **PROGRAM - 1:**

```
//PRG 1
import java.util.*;
class clTravel
{
    int arAge[],arFreqDist[];
    clTravel()
    {
        arAge=new int[10];arFreqDist=new int[5];
    }
    void fnReadAge()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter age of 10 tourists = ");
        for(int i=0;i<10;i++)
            arAge[i]=sc.nextInt();
    }
    void fnFrequency()
    {
        for(int i=0;i<10;i++)
        {
            int a=arAge[i];
            if(a<20)
                arFreqDist[0]++;
            else if(a>20&&a<=40)
                arFreqDist[1]++;
            else if(a>40&&a<=60)
                arFreqDist[2]++;
            else if(a>60&&a<=80)
                arFreqDist[3]++;
            else
                arFreqDist[4]++;
        }
    }
}
```

```

void fnShowFrequency()
{
    System.out.println("Age group\t\tFrequency");
    System.out.println("< 20 \t\t\t"+arFreqDist[0]);
    System.out.println("20 - 40\t\t"+arFreqDist[1]);
    System.out.println("40 - 60\t\t"+arFreqDist[2]);
    System.out.println("60 - 80\t\t"+arFreqDist[3]);
    System.out.println("> 80 \t\t\t"+arFreqDist[4]);
}

void fnShowFreq()
{
    clTravel ob=new clTravel();
    ob.fnReadAge();
    ob.fnFrequency();
    ob.fnShowFrequency();
}

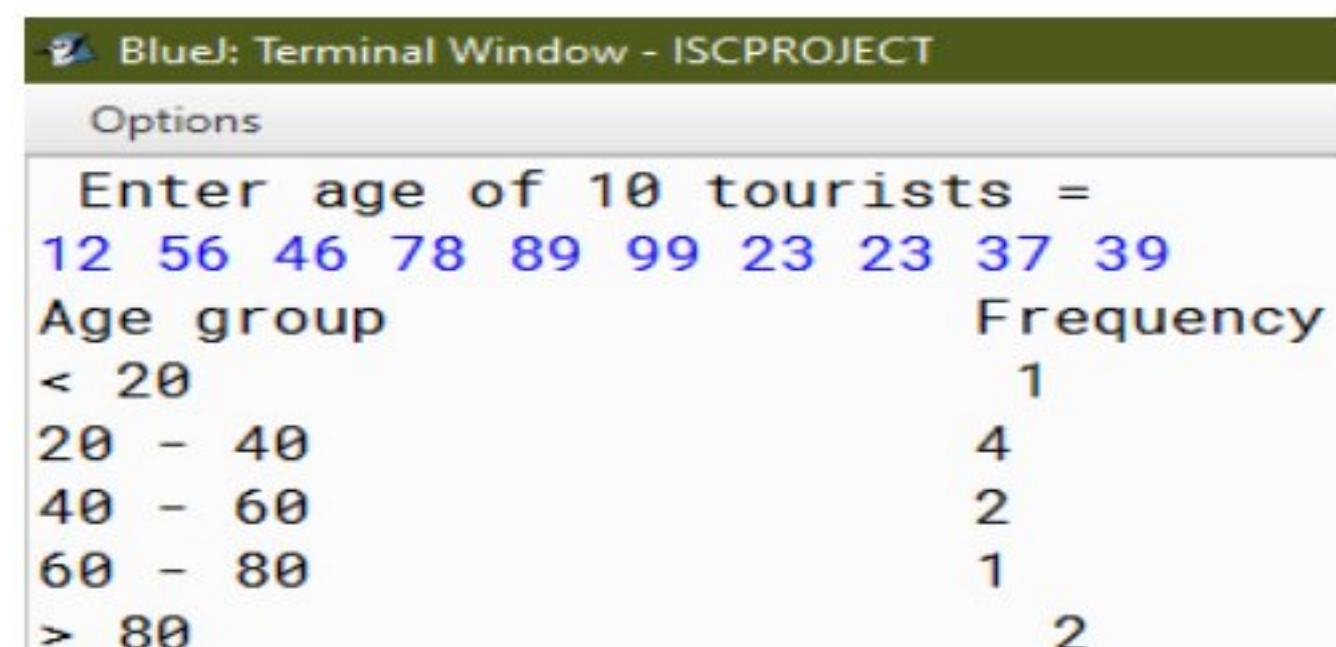
public static void main()
{
    clTravel ob=new clTravel();
    ob.fnReadAge();
    ob.fnFrequency();
    ob.fnShowFrequency();
}
}

```

### Variable Description Table:

Sl.no.	Variable	Datatype	Description
1	ob	clTravel	Object of class to invoke functions.
2	arAge[]	int	Array to store the ages of 10 travellers.
3	arFreqDist[]	int	Array to store the frequency of a particular age group.

### Output1:



BlueJ: Terminal Window - ISCPROJECT

Options

```

Enter age of 10 tourists =
12 56 46 78 89 99 23 23 37 39
Age group          Frequency
< 20              1
20 - 40            4
40 - 60            2
60 - 80            1
> 80              2

```

## **QUESTION 2:**

Write a program in Java that will accept a string from the user that must be comprises of N sentences, where N will be entered by the user and must be greater than 1. Every sentence in the string must bonded with either '?' or '.' or '!'. The string should be a continuous string entered by the user. If the input string does not match with the conditions given above, user should be asked to re-enter the string again. Now ask the user to choose a sentence from the entered string and print that sentence on the screen in Sentence case. Now print a histogram on the frequency of each alphabet present in that sentence with the help of '#' or '\*' only. The histogram should be a column chart on the above-mentioned data.

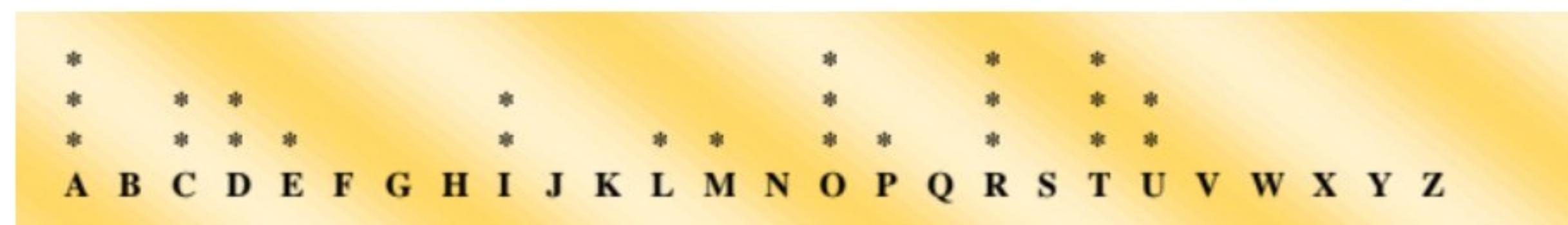
INPUT : Enter the number of sentences for the string : 2

Enter a string having 3 sentences ended with either '?' or '.' or '!'.

Computer Science has theory as well as practical. Today is our computer practical day.

Enter the sentence number of your choice : 2

**OUTPUT :**The sentence of your choice is: Today is our computer practical day.Histogram on the above sentence:



## **SOLUTION - 2:**

### **ALGORITHM:**

1. START
2. Declare a string str to store the user input and also declare ln to store the length of the string. Declare a 2D array mat to store histogram of alphabets and another array alpha to store the alphabets a-z.
3. Read the string from user and store it in str and compute its length in ln.
4. The string is converted to uppercase and its length is stored in a variable ln.
5. Use StringTokenizer to extract each sentence ending with delimiters(. or ? or !).
6. Compute the histogram by comparing the number of each letter of the string with each of the alphabets and storing '\*' in mat counting its frequency.
7. Display the histogram
8. STOP

## **PROGRAM - 2:**

```
//PRG2

import java.util.*;

public class Histogram

{

    // instance variables

    String str;

    int ln;

    char mat[][]; //to store the histogram of the alphabets

    char alpha[]; //to store the alphabets a-z

    // constructor for objects of class histogram

    public Histogram(String s)

    {

        // initialise instance variables

        str = s.toUpperCase();

        ln = str.length();

        alpha = new char[26];

        mat = new char[26][ln];

    }

    public void compute()

    {

        int i=0,j=0;

        for(char a='A'; a<='Z'; a++) //loop to store the alphabets in alpha[]

        {

            alpha[i]=a;

        }

    }

}
```

```

        i++;

    }

    for(i=0; i<26; i++) //total no. of rows as per the total alphabets

    {

        for(int k=0; k<ln; k++) //loop to extract each character from the string

        {

            char c=str.charAt(k);

            if(alpha[i]==c) //alphabet matched

            {

                mat[i][j]='*'; //putting a * in the particular cell ith row & jth

                column

                j++;

            }

        }

        j=0; //for next alphabet j reinitialized to 0

    }

}

public void displayHist()

{

    for(int j=ln-1; j>=0; j--)//column-loop

    {

        for(int i=0; i<26; i++)//row-loop

            System.out.print(mat[i][j]+" ");

        System.out.println();

    }

}

```

```

}

for(int i=0; i<26; i++)
    System.out.print(alpha[i]+" ");
System.out.println();
}

public static void main()
{
    Scanner sc=new Scanner(System.in);
    System.out.println("INPUT :Enter number of sentences for the string (>1):");
    int n=sc.nextInt();
    StringTokenizer st;
    String abc="";
    do
    {
        System.out.println("Enter a string having "+n+" sentences ending with either '?\' or \'.\' or '\!\'");
        abc=sc.nextLine();
        st=new StringTokenizer(abc,".?!");
    }while(st.countTokens()!=n);
    System.out.println("Enter the sentence number of your choice");
    n=sc.nextInt();
    for(int i=1;i<=n;i++)
        abc=st.nextToken();
    Histogram ob=new Histogram(abc);
    ob.compute();
}

```

```

        ob.displayHist();
    }

}

```

### Variable Description Table -2

Sl.No.	Variable	Datatype	Description
1	sc	Scanner	Object of Scanner class to take user input.
2	n	int	Stores input
3	st	StringTokenizer	Object to extract each sentence ending with delimiters(. or ? or !).
4	abc	String	Takes n sentences as input
5	i	int	Loop control variable
6	str	String	Stores s abc in upper case.
7	ln	int	Stores string length
8	mat[][]	char	Stores the histogram
9	alpha[][]	char	Stores all the alphabets

### OUTPUT - 2:

```

BlueJ: Terminal Window - ISCPROJECT
Options
INPUT :Enter number of sentences for the string (>1):
2
Enter a string having 2 sentences ending with either '?' or '.' or '!'
Enter a string having 2 sentences ending with either '?' or '.' or '!'
Computer Science has theory as well as practical.Today is our computer practical day.
Enter the sentence number of your choice2

*
*   *           *   *   *
*   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

```

### **QUESTION 3:**

Input a sentence from the user and change the vowels to next alphabet and the consonant to previous alphabet of that word in that string. Display both the input and changed string. Class description is given below:-

Class name : WordChange

Data members/instance variable

String s : input string

String r : resultant string

Member functions:-

WordChange(String) parameterized constructor to initialize the data members accordingly

String change(String) return the changed word as instructed in the question

void extract(void) extract each word from the string s and pass to the change() and form the new string.

void display() display both the strings

Specify the class WordChange giving the details of constructor and all the functions mentioned in the

question. You need to write the main() method and create an object of the class and call the functions accordingly.

### **SOLUTION – 3**

#### **ALGORITHM:**

1. START
2. Takes a string as user input
3. Extracts each word from the string
4. Changes each word according to the given conditions while extracting
5. Displays the resultant string
6. STOP

### **PROGRAM - 3:**

//PRG 3

```
import java.util.*;
class WordChange
{
    String s,r;
    WordChange(String a)
    {
        s=a;r="";
    }

    String change(String a)
    {
        String abc="";
        for(int i=0;i<a.length();i++)
        {
            abc="";
            char c=a.charAt(i);

            if(c=='A'||c=='E'||c=='I'||c=='O'||c=='U'||c=='a'||c=='e'||c=='i'||c=='o'||c=='u')
            {
                c+=1;abc+=c;
            }
            else
            {
                c-=1;abc+=c;
            }
        }
        return abc;
    }

    void extract()
    {
        StringTokenizer st=new StringTokenizer(s);
        r+=change(st.nextToken());
    }

    void display()
    {
        System.out.println("Input string = "+s+"\nResultant string = "+r);
    }
}
```

```

public static void main()
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the string = ");
    String str=sc.nextLine();
    WordChange ob =new WordChange(str);
    ob.extract();
    ob.display();
}

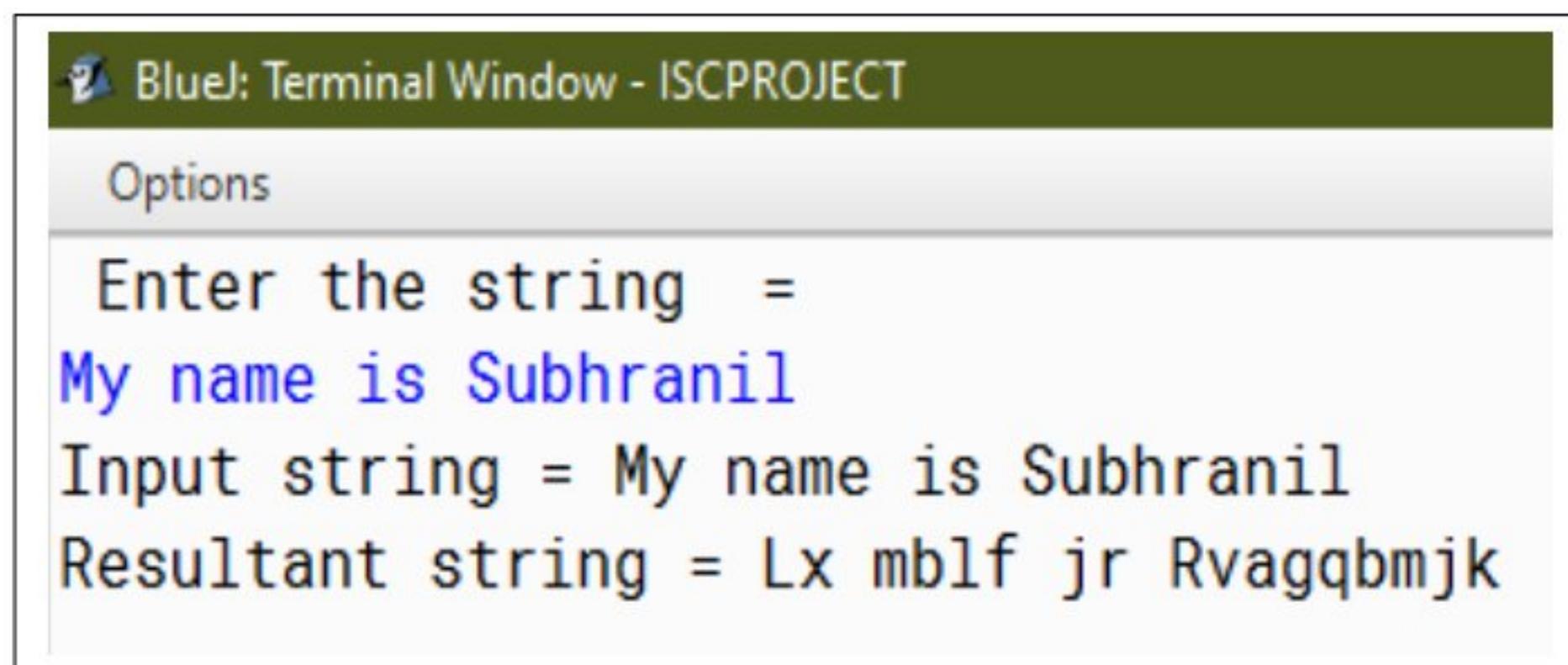
}

```

### **Variable Description Table:**

<b>Sl.no.</b>	<b>Variable</b>	<b>Datatype</b>	<b>Description</b>
1	sc	Scanner	Object of Scanner class to take user input.
2	str	String	Stores string input
3	ob	WordChange	Object of the class to invoke functions.
4	s	String	Stores str
5	r	String	Stores resultant string

### **OUTPUT:**



```

BlueJ: Terminal Window - ISCPROJECT
Options
Enter the string =
My name is Subhranil
Input string = My name is Subhranil
Resultant string = Lx mbif jr Rvaggbmjk

```

#### **QUESTION 4 :**

A tourist company plans to organize tour to visit N major cities of India. WAP to accept minimum name of 4 cities and the starting location from the user. Now print the possible combinations to travel all those cities so that each city is to be routed only once.

#### **SOLUTION:**

##### **Algorithm: -**

- \* Step 1: -Start.
- \* Step 2: -Take no. of cities to visit from user.
- \* Step 3: -Take all the city names.
- \* Step 4: -Leaving starting city permute rest of cities in all possible ways.
- \* Step 5: -Print Routes Found.
- \* Step 6: -End.

## PROGRAM 4:

### Program: -

```
import java.util.*;
class Routes
{
    int n;// To store the number of cities
    int rn;// To store the route number
    String cities[];// To store the cities
    String start;//To store the starting city
    Routes()
    {
        n = 0;//Initializing default values.
        rn = 1;
        start = "";
    }//end of Routes
    // Function to validate and get the number of cities
    void get_number(String s)
    {
        Scanner sc=new Scanner(System.in);
        System.out.print(s);
        try
        {
            n = sc.nextInt();
            if(n < 4) // Validating input
            {
                System.out.println("At least 4 cities are required");
                get_number(s); // Calling the function again when user entered an invalid input
            }
        }
        catch(Exception e) // Handeling error when user enters something other than an
        integer
        {
            System.out.println("\nPlease enter an integer only");
            get_number(s); // Calling the function again when user entered an invalid input
        }
        cities = new String[n]; // Initializing the cities array
    }
    // Function to validate and get the names of cities from the user
    void get_cities(String str)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println(str);// Taking input
        String s = sc.nextLine();
        s = s.trim(); // Removing extra spaces
        int l = s.length();
        String ns=""; // temporary variable
        int a = 0, e = 0;
```

```

for(int i = 0; i < l; i++)
{
    char ch = s.charAt(i);
    if(ch==',' || ch=='')
    {
        try
        {
            cities[a] = ns.trim(); // Placing the name of the city into the cities
            if(ch==';')
                break; // breaking from loop when semi colon detected
            a++;
        }
        catch(ArrayIndexOutOfBoundsException ex) // catching error when user enters
more cities
        {
            System.out.println("Please only enter as many cities as you specified");
            e = 1;
            break;
        }
        ns = "";
    }
    else if(a<n && i==l-1) // Checking if the user entered less number of cities
    {
        System.out.println("Please enter as many cities as you specified");
        e = 1;
        break;
    }
    else
    {
        if(i == l-1) // Checking if the user ended the list with a semi colon or not
        {
            System.out.println("Please make sure you placed a semi-colon(;) at the end");
            e = 1;
            break;
        }

        else
        ns = ns + ch;
    }
}
if(cities[n-1]==null && e==0) // Final checking if the user entered less number fo cities
{
    System.out.println("Please enter as many cities as you specified");
    get_cities(str);
}
else if(e!=0) // if there was any error
get_cities(str); // calling the function again
}
// Function to get the name of the starting sity

```

```

void get_starting_city()
{
    // Taking input
    Scanner sc=new Scanner(System.in);
    System.out.print("Enter the city you want to start with : ");
    start = sc.nextLine().trim(); // trimming extra spaces
    //Checking if the name of the city exists in the list entered
    boolean found = false; // setting found as false
    int i = 0; // LCV
    while(i < n)
    {
        if(found==true)
            break; // breaking if already found
        if(cities[i].equals(start))
            found=true; // setting found as true if found
        i++;
    }
    if(found)
    {
        // if found, removing the city from the array to make things easy
        String newarr[] = new String[n-1]; // initialising a temporary array with size 1 less than
        the original array
        int a = 0;
        for(int j = 0; j < n; j++)
        {
            if(j==i-1)
                continue; // continuing if the city is the starting city
            else
            {
                newarr[a] = cities[j];
                a++;
            }
        }
        cities = new String[n-1]; // re-initialising the original array with size 1 less than before
        for(int j = 0; j < n-1; j++)
    }

    cities[j] = newarr[j]; // re-filling the array using the temporary array we just filled
    return;
}
else
{
    System.out.println("The city was not found in the list");
    get_starting_city(); // If the city was not found re-running the function
}
}

```

```

// Displaying the possible routes
void display(String s1[], String s2[])
{
    if (s2.length <= 1)
    {
        int l = s1.length + s2.length;
        int l1 = s1.length;
        System.out.print("Route " + rn + ": " + start + " =>"); // Printing the route generated
        for(int i = 0; i < l; i++)
        {
            // The if-else block is for printing the route generated
            if(i < l1)
            {
                if(i<l-1)
                    System.out.print(s1[i] + " => ");
                else
                    System.out.println(s1[i]);
            }
            else
            {
                if(i == l-1)
                    System.out.println(s2[i-l1]);
                else
                    System.out.print(s2[i-l1] + " => ");
            }
        }
        rn++; // incrementing the route number
    }
    else
    {
        // Loop to generate routes
        for (int i = 0; i < s2.length; i++)
        {
            String x[]=new String[1]; // Temporary array to store the ith element of s2 array
            x[0] = s2[i]; // initialising the only term in the array to the ith element in the s2
            array
            String y[]=new String[i]; // Temporary array to store all the elements before the ith
            element in the s2 array
            for(int j = 0; j < i; j++)
                y[j] = s2[j]; // initialising the elements of the array
            String z[]=new String[s2.length - (i+1)]; // Temporary array to store all elements in
            s2 array after the ith element
            for(int j = 0; j < z.length; j++)

```

```

        z[j] = s2[i+j+1]; // initialising the elements of the array
        String p[] = new String[s1.length + x.length]; // Temporary array to store the
        elements of s1 and x arrays respectively
        for(int j = 0; j < p.length; j++) // Filling the array p
            if(j < s1.length)
                p[j] = s1[j];
            else
                p[j] = x[j-s1.length];
        String q[] = new String[y.length + z.length]; // Temporary array to store the elements
        of y and z arrays respectively
        for(int j = 0; j < q.length; j++) // Filling the array q
            if(j < y.length)
                q[j] = y[j];
            else
                q[j] = z[j-y.length];
        display(p, q); // Calling the function with p and q as the parameters
    }
}
}
// Function that calls the display function because display function is recursive
void displayRoutes()
{
    System.out.println("Possible routes are :-");
    String x[] = new String[0]; // initialising a null array
    display(x, cities); // calling the display function with null array and cities array
    respectively
}
// Main method to create object and call the required functions
public static void main(String[] args)
{
    Routes rts = new Routes(); // Creating Object.
    rts.get_number("Enter the number of cities you want to travel to : ");
    rts.get_cities("Enter the names of the cities SEPERATED BY COMMAS(,) and END THE
LIST BY A SEMICOLON(;)");
    rts.get_starting_city(); // Calling functions.
    rts.displayRoutes();
} // end of main
} // end of class

```

#### Output Screen:-

 BlueJ Terminal Window - Shivang  
 Options  
 Enter the number of cities you want to travel to : 4  
 Enter the names of the cities SEPERATED BY COMMAS(,) and END THE LIST BY A SEMICOLON(;)  
 Bangalore,Delhi,Mumbai,Kolkata;  
 Enter the city you want to start with : Kolkata  
 Possible routes are :-  
 Route 1: Kolkata => Bangalore => Delhi => Mumbai  
 Route 2: Kolkata => Bangalore => Mumbai => Delhi  
 Route 3: Kolkata => Delhi => Bangalore => Mumbai  
 Route 4: Kolkata => Delhi => Mumbai => Bangalore  
 Route 5: Kolkata => Mumbai => Bangalore => Delhi  
 Route 6: Kolkata => Mumbai => Delhi => Bangalore

**Variable Description Chart: -**

Sl.No.	Datatype	Variable	Description/Usage
1	Scanner	sc	Object of Scanner Class.
2	int	n	To store the number of cities.
3	int	rn	To store the route number.
4	String[]	cities	To store the cities .
5	String	start	To store the starting city.
6	String	s	To take inputs.
7	int	l	To store length of string.
8	String	ns	Temporary variable.
9	int	a	To count no. of iterations.
10	int	e	To check for errors.
11	char	ch	To extract cities.
12	int	i	To run loop.
13	boolean	found	Checking if name of the city is in list.
14	int	j	To run loop.
15	String[]	x	To store Routes generated.
16	String[]	s1	To store value of x passed.
17	String[]	s2	To store cities passed.
18	int	l	To store length of sum of s1 and s2.
19	int	l1	To store length of s1.
20	String[]	x	Array to store the ith element of s2 array.
21	String[]	y	Array to store elements before the ith element in the s2 array.
22	String[]	z	Array to store elements in s2 array after the ith element.
23	String[]	p	Array to store the elements of s1 and x arrays respectively.
24	String[]	q	Array to store the elements of y and z arrays respectively.

### **QUESTION 5:**

Anagram of a word is all the possible combination of alphabets present in that particular word.

Write a program in Java to accept one single word of any length from the user and print the anagrams of that word. The program should check that the word should consist of alphabets and can be of any length. Also print the total number of words displayed.

### **SOLUTION:**

#### **ALGORITHM:**

1. START
2. Takes the word as user input.
3. Calls show() function. Follows a recursive logic inside an iterative block to print all the anagrams.
4. Calls the factorial function() to print the number of anagram words generated.
5. STOP

## **PROGRAM-5:**

```
//PRG 5
import java.util.*;
class Anagram
{
    int cr;
    Anagram()
    {
        cr=0;
    }
    public void show(String s,String inp)
    {
        if(inp.length()<=1)
        {
            cr++;
            System.out.print(s+inp+"\t");
            if(cr%10==0)
                System.out.println();
        }
        else
        {
            for(int i=0;i<inp.length();i++)
            {
                String t1=inp.substring(i,i+1);
                String t2=inp.substring(0,i);
                String t3=inp.substring(i+1);
                show(s+t1,t2+t3);
            }
        }
    }
    int factorial(int t)
    {
        if(t==1)
            return 1;
        else
            return factorial(t-1)*t;
    }
    public static void main()
```

```

{
    Anagram ob=new Anagram();
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter a word = ");
    String w=sc.next().trim();
    System.out.println("Anagrams of the word "+w+" are - ");
    ob.show("",w);
    System.out.println("\nTotal number of words =
"+ob.factorial(w.length()));
}
}

```

### Variable Description Table:

Sl.no.	Variable	Datatype	Description
1	ob	Anagram	Object of Anagram class to invoke member functions.
2	sc	Scanner	Object of Scanner class to take user input
3	w	String	Takes string input
4	cr	int	Used to generate anagram words.

### OUTPUT:

```

BlueJ: Terminal Window - ISCPROJECT
Options
Enter a word =
WORD
Anagrams of the word WORD are -
WORD    WODR    WROD    WRDO    WDOR    WDRD    OWRD    OWDR    ORWD    ORDW
ODWR    ODRW    RWOD    RWDO    ROWD    RODW    RDWO    RDOW    DWOR    DWRO
DOWR    DORW    DRWO    DROW
Total number of words = 24

```

## QUESTION 6

A prime number is a number that is divisible by 1 and that number.

Twin prime numbers are the pair of

2 prime numbers whose difference is 2, e.g (3,5), (5,7), (11,13) etc. The sum of reciprocals of the twin

primes converges to a sum, known as Brun's Constant i.e.

$$\left(\frac{1}{3} + \frac{1}{5}\right) + \left(\frac{1}{5} + \frac{1}{7}\right) + \left(\frac{1}{11} + \frac{1}{13}\right) + \dots \text{upto } n^{\text{th}} \text{ term}$$

Declare a class named "Primes" with one data member double sum and three member functions int

primeCheck( int, int ), double sumTwinPrime( int ) and void BrunConstant ( int ).

WAP to declare the above class with its member functions. Use recursive technique in primeCheck( )

function. Write the main method also.

SOLUTION:

Algorithm:

1. START
2. Taking number of terms as input.
3. Calling BrunConstant() function which in turn calls sumTwinPrime() function which generates the series with the help of primeCheck() recursive function.
4. STOP.

## PROGRAM-6:

```
//PRG 6
import java.util.*;
class Primes
{
    double sum;
    Primes()
    {
        sum=0.0;
    }
    int primeCheck(int a,int b)
    {
        if(b>Math.sqrt(a))
            return 1;
        else
        {
            if(a%b==0)
                return 0;
            else
                return primeCheck(a,b+1);
        }
    }
    double sumTwinPrime(int a)
    {
        int c=0;
        System.out.println("The series :");
        for(int i=3;c<a;i+=2)
        {
            if(primeCheck(i,2)==1 && primeCheck(i+2,2)==1)
            {
                sum+=(1.0/i+1.0/(i+2));
                c++;
                System.out.print("+(1/" + i + "+1/" + (i+2) + ")");
            }
        }
        System.out.println();
        return sum;
    }
}
```

```

}

void BrunConstant(int a)
{
    System.out.println("The sum = "+sumTwinPrime(a));
}

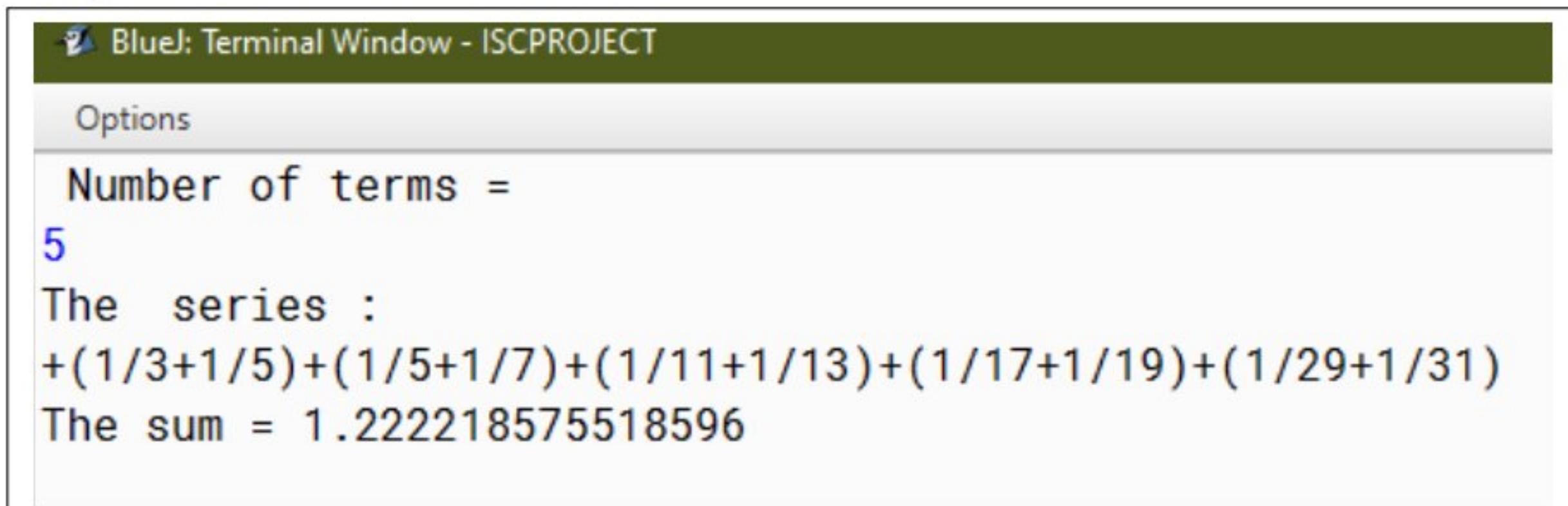
public static void main()
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Number of terms = ");
    int n=sc.nextInt();
    Primes ob=new Primes();
    ob.BrunConstant(n);
}
}

```

### Variable Description Chart:

Sl.no.	Variable	Datatype	Description
1	sc	Scanner	Object of Scanner class to take user input
2	n	int	Stores integer user input
3	ob	Primes	Object of Primes class to invoke functions.
4	sum	double	Stores the resulting sum of series.

### Output:



The screenshot shows a terminal window titled "BlueJ: Terminal Window - ISCPROJECT". The window displays the following output:

```

Options
Number of terms =
5
The series :
+(1/3+1/5)+(1/5+1/7)+(1/11+1/13)+(1/17+1/19)+(1/29+1/31)
The sum = 1.222218575518596

```

## **QUESTION 7:**

Write a program to perform the following task:

Input a number, say mak. Now create a number, say tip, by arranging the digits of the number in ascending order. Create another number, say tap, by arranging the digits of the number in descending order. List all the Perfect Squares that lie in between tip and tap. [A perfect square is a number that has an integer square root.]

## **SOLUTION**

### **Algorithm:**

1. START
2. Takes a number as user input
3. Counts the digits of the number and stores in c
4. Declares the size of A1[] and A2[] as c
5. Stores the digits of the no. in A1[] and A2[]
6. Sorts A1[] in ascending order and A2[] in descending order
7. Converts the digits in A1[] and A2[] back into numbers tip and tap
8. Prints all the perfect squares between tip and tap
9. STOP

## **Program – 7:**

```
//PRG 7
import java.util.*;
class PSquare
{
    int mak,tip,tap,A1[],A2[];

    PSquare(int m,int c)
    {
        mak=m;A1=new int[c];A2=new int[c];tip=o;tap=o;
    }

    void toArray()//Stores digits of mak in A1[] and A2[].
    {
        int t=mak;int c=A1.length-1;int
        rem;//System.out.println(mak);
        while(t!=o)
        {
            rem=t%10;
            A1[c]=rem;A2[c]=rem;
            t/=10;
            c--;
        }
    }

    void sort()
    {
        int t;
        for(int i=o;i<A1.length;i++)
        {

            for(int j=o;j<A1.length-i-1;j++)
            {
                if(A1[j]>A1[j+1])
                {
                    t=A1[j];A1[j]=A1[j+1];A1[j+1]=t;
                    System.out.println(A1[j]+\t"+A1[j+1]+" within A1
loop");
                }
            }
        }
    }
}
```

```

}

    if(A2[j]<A2[j+1])
    {
        t=A2[j];A2[j]=A2[j+1];A2[j+1]=t;
        System.out.println(A2[j]+\t"+A2[j+1]+" within A2 loop");
    }
}

for(int i=0;i<A1.length;i++)
{
    System.out.print(A1[i]+\t");System.out.println(A2[i]);
}
}

void fromArray()
{
    for(int i=0;i<A1.length;i++)
    {
        tip=tip*10+A1[i];
        tap=tap*10+A2[i];
    }

    System.out.println("New number 1 = "+tip);
    System.out.println("New number 2 = "+tap);
}

void perfect()
{
    System.out.println("The desired list is = ");
    int a=(int)Math.sqrt(tip)+1;
    int b=(int)Math.sqrt(tap)-1;
    for(int i=a;i<=b;i++)
    {
        System.out.print((i*i)+"\t");
    }
}

```

```

public static void main()
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter Number = ");
    int mak=sc.nextInt();
    int t=mak;int c=0;
    while(t!=0)
    {
        t/=10;c++;
    }
    PSquare ob=new PSquare(mak,c);
    ob.toArray();
    ob.sort();
    ob.fromArray();
    ob.perfect();
}
}

```

### **Variable Description Table:**

<b>Sl.no.</b>	<b>Variable</b>	<b>Datatype</b>	<b>Description</b>
1.	sc	Scanner	Object of Scanner class to take user input.
2.	mak	int	Stores integer user input.
3	c	int	Stores the no. of digits in mak.
4	tip	int	Stores digits of mak in ascending order.
5	tap	int	Stores digits of mak in descending order.
6	A1[]	int	Array to sort digits of mak in ascending order.
7	A2[]	int	Array to sort digits of mak in descending order.

## **Output 7:**

```
BlueJ Terminal Window - ISCPROJECT
Options
Enter Number =
302
New number 1 = 23
New number 2 = 320
The desired list is =
25      36      49      64      81      100      121      144      169      196      225      256
```

### **QUESTION 8:**

Given the following series:

$$S = m! / pm + (m-1)! / pm-1 + (m-2)! / pm-2 + \dots + (m-m)! / pm-m$$

A class called `clSomeSeries` has been defined to calculate the sum of the series. Some of the function/methods in `clSomeSeries` are shown below:

Class name	<b>clSomeSeries</b>
Data members:	int m, p double sum
Member functions/methods: <b>NewSeries ( )</b>	Constructor
<code>int fnFact (int num)</code>	Calculates and returns the factorial of num (num!) where $num! = 1*2*3*\dots*num$
<code>long fnPower(int a, int p)</code>	Calculates and returns the value of $a^p$ , without using header file <code>&lt;math.h&gt;</code>
<code>void fnCalculate (void)</code>	Calculates the sum of the given series in "sum" using other member functions
<code>void fnInput (void)</code>	Inputs the value of m, p.
<code>void fnPrint(void)</code>	Shows the sum of the series.

Specify the class `clSomeSeries`, giving details of constructor( ) and mentioned functions. Write the main method also.

### **Algorithm - 8:**

1. START
2. Takes the value of m and p as user input.
3. Calculates the sum of series
4. Prints the sum.
5. STOP

### **Program 8:**

```
//PRG 8
import java.util.*;
class clSomeSeries
{
    int m,p;double sum;
    clSomeSeries()
    {
        sum=0;
    }
    int fnFact(int num)
    {
        if(num==0)
            return 1;
        else
            return fnFact(num-1)*num;
    }
    long fnPower(int a,int p)
    {
        if(p==0)
            return 1;
        else
            return a*fnPower(a,p-1);
    }
    void fnCalculate()
    {
        double t;
        for(int i=0;i<=m;i++)
        {
            t=fnFact(m-i);
            sum+=t/fnPower(p,m-i);

        }
    }
    void fnPrint()
    {
        System.out.println("Sum of series = "+sum);
    }
}
```

```

void fnInput()
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the values of m and p = ");
    m=sc.nextInt();
    p=sc.nextInt();
}
public static void main()
{
    clSomeSeries ob =new clSomeSeries();
    ob.fnInput();
    ob.fnCalculate();
    ob.fnPrint();
}
}

```

### **Variable Description Table:**

<b>Sl.no.</b>	<b>Variable</b>	<b>Datatype</b>	<b>Description</b>
1.	ob	clSomeSeries	Object of class to invoke functions.
2	m	int	Variable used in series.
3	p	Int	Variable used in series.
4	sum	double	Stores the resultant sum of series

### **Output 8:**

```

BlueJ: Terminal Window - ISCPROJECT
Options
Enter the values of m and p =
3
4
Sum of series = 1.46875

```

### **QUESTION 9:**

Write a program to generate a 3X3 magic square i.e. a square in which the sum of all the

numbers in each row, in each column. Now display the sum along with the array elements as a

single unit. Finally perform :

- (i) Row-wise sorting in ascending order
- (ii) Column-wise sorting in descending order

### **SOLUTION:**

#### **Algorithm:9**

1. START
2. Generates the magic square in generateSquare() function and the following logic:

In any magic square, the first number i.e. 1 is stored at position  $(n/2, n-1)$ . Let this position be  $(i,j)$ . The next number is stored at position  $(i-1, j+1)$  where we can consider each row & column as circular array i.e. they wrap around.

Three conditions hold:

  - \* The position of next number is calculated by decrementing row number of the previous number by 1, and incrementing the column number of the previous number by 1. At any time, if the calculated row position becomes -1, it will wrap around to  $n-1$ . Similarly, if the calculated column position becomes  $n$ , it will wrap around to 0.
  - \* If the magic square already contains a number at the calculated position, calculated column position will be decremented by 2, and calculated row position will be incremented by 1.
  - \* If the calculated row position is -1 & calculated column position is  $n$ , the new position would be:  $(0, n-2)$ .
3. Sorts the rows in ascending order and prints.
4. Sorts the columns in descending order and prints.
5. STOP.

### **Program 9:**

```
//PRG 9
import java.io.*;

class MagicSquare
{
    int magicSquare[][][],RSort[][][],CSort[][];int n;
    MagicSquare(int n)
    {
        magicSquare=new int[n][n];RSort=new int[n][n];CSort=new
        int[n][n];this.n=n;
    }

    // Function to generate odd sized magic squares
    void generateSquare()
    {

        // Initialize position for 1
        int i = n / 2;
        int j = n - 1;

        // One by one put all values in magic square
        for (int num = 1; num <= n * n;) {
            if (i == -1 && j == n) // 3rd condition
            {
                j = n - 2;
                i = 0;
            }
            else {
                // 1st condition helper if next number
                // goes to out of square's right side
                if (j == n)
                    j = 0;

                // 1st condition helper if next number is
                // goes to out of square's upper side
                if (i < 0)
                    i = n - 1;
```

```

}

// 2nd condition
if (magicSquare[i][j] != 0) {
    j -= 2;
    i++;
    continue;
}
else
// set number
    magicSquare[i][j] = num++;

// 1st condition
j++;
i--;
}

// print magic square
System.out.println("The Magic Square for " + n
+ ":");

System.out.println("Sum of each row or column "
+ n * (n * n + 1) / 2 + ":");

for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++)
        System.out.print(magicSquare[i][j] + " ");
    System.out.println();
}
}

void initializeSort()
{
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            RSort[i][j]=magicSquare[i][j];
            CSort[i][j]=magicSquare[i][j];
        }
    }
}

```

```

}

}

void Sort()
{
    int temp;
    for(int i=0;i<n;i++)
    {

        for(int j=0;j<n ;j++)
        {
            for(int k=0;k<n-j-1;k++)
            {
                if(RSort[i][k]>RSort[i][k+1])
                {
                    temp =
RSort[i][k];RSort[i][k]=RSort[i][k+1];RSort[i][k+1]=temp;

                }

                if(CSort[k][i]<CSort[k+1][i])
                {
                    temp =
CSort[k][i];CSort[k][i]=CSort[k+1][i];CSort[k+1][i]=temp;
                }
            }
        }
    }
}

System.out.println("After sorting each row of Magic Square in
Ascending order :");
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
        System.out.print(RSort[i][j] + " ");
    System.out.println();
}
System.out.println("After sorting each column of Magic Square
in Descending order :");

```

```
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
        System.out.print(CSort[i][j] + " ");
    System.out.println();
}

public static void main(String[] args)
{
    // Works only when n is odd
    MagicSquare ob =new MagicSquare(3);
    ob.generateSquare();
    ob.initializeSort();

    ob.Sort();
}

}
```

### Variable Description Table:

Sl.no.	Variable	Datatype	Description
1	ob	MagicSquare	Object of class to invoke functions.
2	magicSquare[][]	int	Stores the 3*3 magic square
3	RSort[][]	Int	Stores the magic square by sorting each of its rows in ascending order.
4	CSort[][]	int	Stores the magic square by sorting each of its columns in descending order
5	n	int	Stores the side length(3) of magic square.

### Output 9:

```

BlueJ: Terminal Window - ISCPROJECT
Options
The Magic Square for 3:
Sum of each row or column 15:
2 7 6
9 5 1
4 3 8
After sorting each row of Magic Square in Ascending order :
2 6 7
1 5 9
3 4 8
After sorting each column of Magic Square in Descending order :
9 7 8
4 5 6
2 3 1

```

### **QUESTION 10:**

Write a program to implement the following:

In a private detective department, the trainee detectives were given certain rules for making their

passwords. A password would be considered valid only if

- i. It had odd number of characters less than 10.
- ii. The characters would be alternate alphabets and digits.
- iii. The alphabets would be only between J and T (both inclusive)
- iv. No even digit could be present after character J or T.

### **SOLUTION10:**

#### **Algorithm:**

1. START
2. Takes the password input.
3. Checks for the given conditions to verify the validity of the password and prints respectively.
4. STOP

### **Program 10:**

```
//PRG 10
import java.util.*;
class Pass
{
    String p;
    Pass(String P)
    {
        p=P;
    }

    boolean checkP()
    {

        if(p.length()>=10)
            return false;

        p=p.toUpperCase();
        char a;boolean f;
        for(int i=0;i<p.length()-1;i++)
        {
            a=p.charAt(i);f=false;
            if(a=='J'||a=='T')
                f=true;

            if(!((a>='J'&& a<='T')||(a>=48 && a<=57)))
                return false;

            if(a>=48 && a<=57)
            {
                a=p.charAt(i+1);
                if(!(a>='J'&& a<='T'))
                    return false;
            }
        }
    }
}
```

```

else
{
    a=p.charAt(i+1);
    if(f)
    {
        if(a==48||a==50||a==52||a==54||a==56)
            return false;
    }
    if(!(a>=48&&a<=57))
        return false;
}

return true;
}

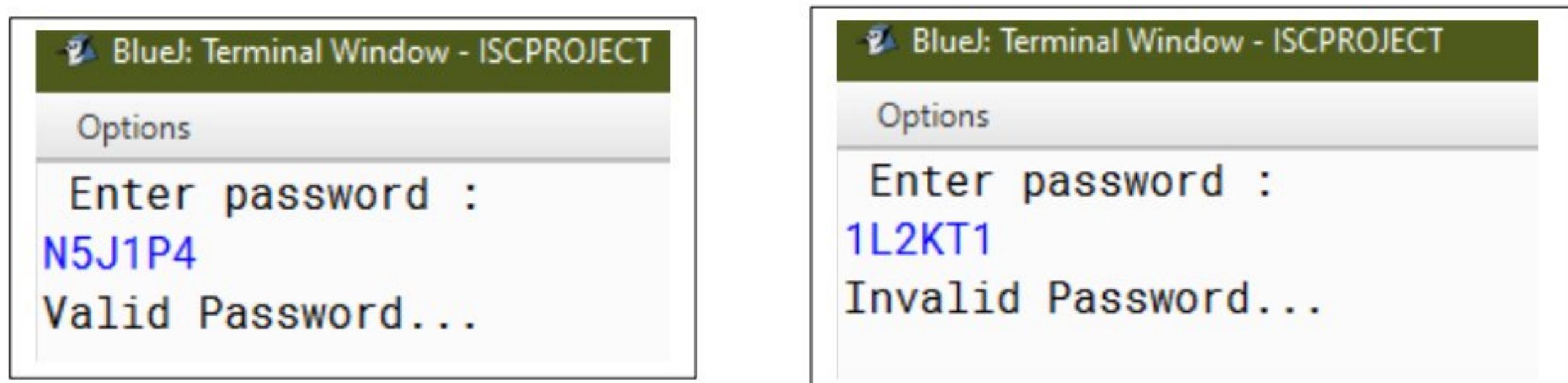
public static void main()
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter password :");
    String p=sc.nextLine();
    Pass ob=new Pass(p.trim());
    if(ob.checkP())
        System.out.println("Valid Password...");
    else
        System.out.println("Invalid Password...");
}
}

```

### Variable Description Table:

Sl.No.	Variable	Datatype	Description
1	sc	Scanner	Object of Scanner class to take user input.
2	p	String	Stores the password input.
3	ob	Pass	Object of class to invoke functions.

### Output10:



The image contains two side-by-side screenshots of a BlueJ terminal window titled "BlueJ: Terminal Window - ISCPROJECT". Both windows show the same interface with a menu bar and a text area. In the first window, the text area displays "Enter password :" followed by the input "N5J1P4" in blue, and "Valid Password..." below it. In the second window, the text area displays "Enter password :" followed by the input "1L2KT1" in blue, and "Invalid Password..." below it.

### **QUESTION 11:**

In Bello Labs, a group of researchers were working on a string balancing project. Their team leader declared that a string would be called “Well Balanced String” if it satisfied the following conditions –

- i. No character other than alphabet ‘a’ , ‘z’ would be present in the string
- ii. against every character ‘a’, character ‘z’ would be present
- iii. a pair of ‘a’ and ‘z’ would be such that in totality, no ‘z’ appears without having a preceding ‘a’ for it.

### **SOLUTION11:**

#### **Algorithm11:**

1. START
2. Takes the string user input
3. Checks whether the string is balanced or not based on the given conditions and prints respectively.
4. STOP.

## **Program11:**

```
//PRG 11
import java.util.*;
class Balanced
{
    String str;
    Balanced(String s)
    {
        str = s;
    }

    boolean check()
    {
        int count=0;char c;
        for(int i=0;i<str.length();i++)
        {
            c=str.charAt(i);
            if(c!='a' && c!='z')
                return false;
            if(c=='a')
                count++;
            else
                count--;
            if(count<0)
                return false;
        }
        if(count!=0)
            return false;
        return true;
    }

    public static void main()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the string = ");
        String str=sc.next();
        Balanced ob=new Balanced(str);
```

```

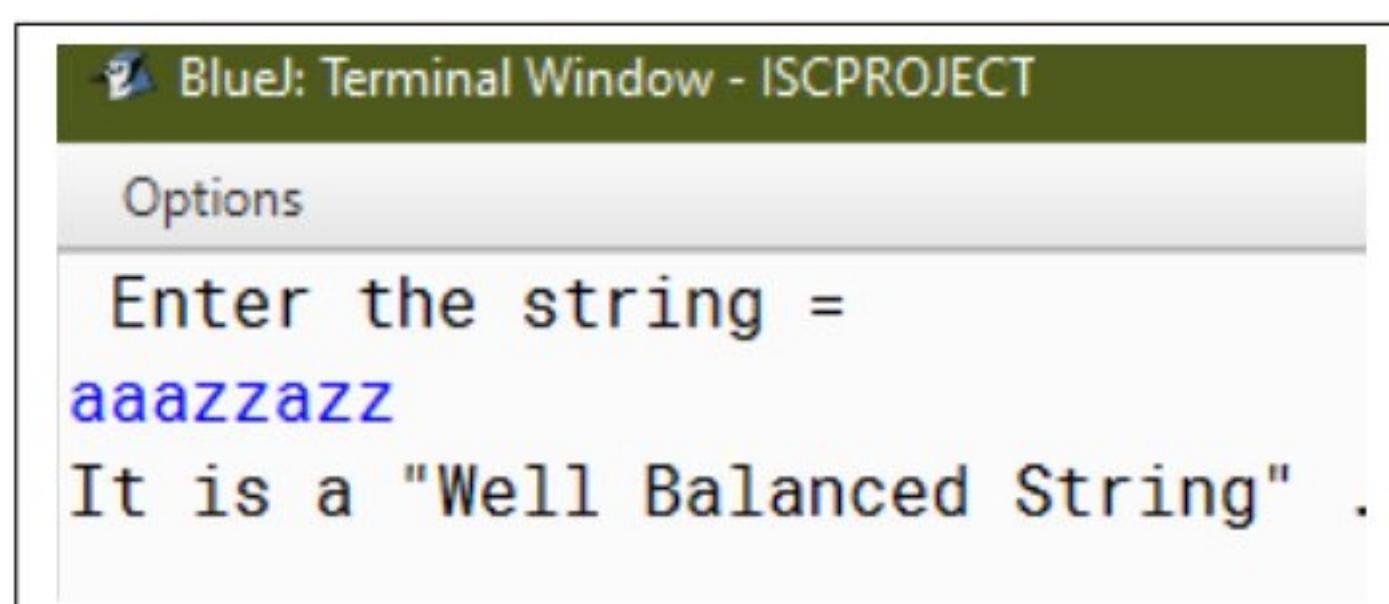
if(ob.check())
    System.out.println("It is a \"Well Balanced String\" .");
else
    System.out.println("It is NOT a \"Well Balanced String\" "
    .);
}
}

```

### **Variable Description Table11:**

<b>Sl.No.</b>	<b>Variable</b>	<b>Datatype</b>	<b>Description</b>
1	sc	Scanner	Object of Scanner class to take user input
2	str	String	Stores the string user input
3	ob	Balanced	Object of class to invoke functions.

### **Output11:**

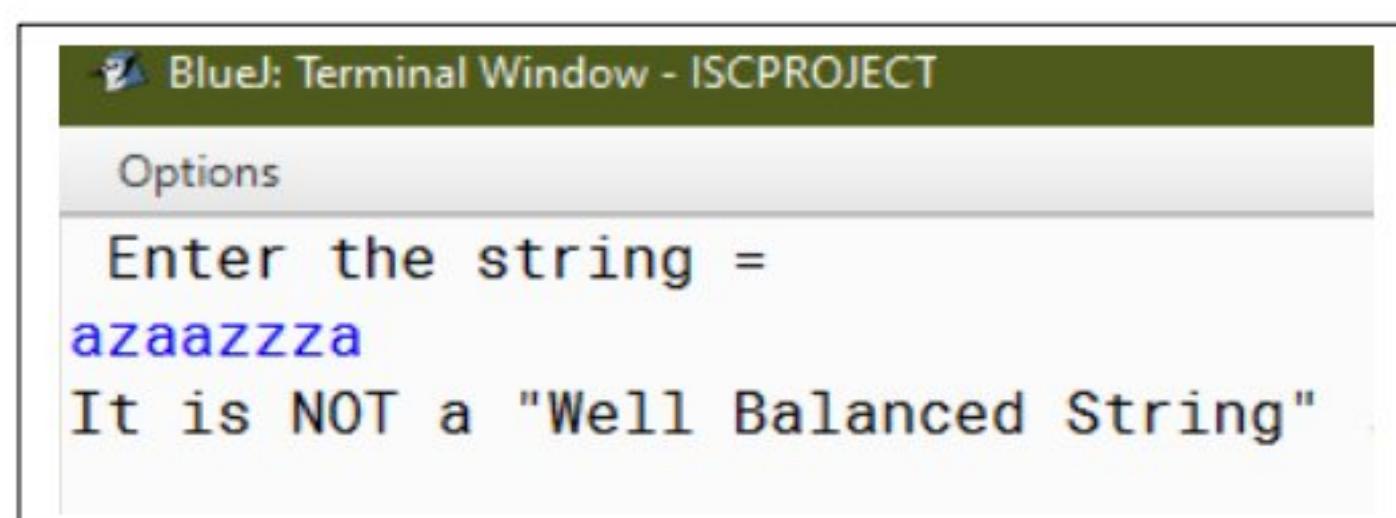


BlueJ: Terminal Window - ISCPROJECT

Options

Enter the string = aaazzazz

It is a "Well Balanced String" .



BlueJ: Terminal Window - ISCPROJECT

Options

Enter the string = azaazzza

It is NOT a "Well Balanced String" .

### **QUESTION 12:**

Write a program to convert a roman number to decimal equivalent and vice-versa.

### **SOLUTION 12:**

#### **Algorithm12:**

1. START
2. Takes the input as roman or decimal depending on user's choice.
3. Converts the input to the other form .
4. Prints the respective other form.
5. STOP.

## **Program12:**

```
//PRG 12
import java.util.*;
class Convert
{
    void toRoman(int n)
    {
        int A[]={1,4,5,9,10,40,50,90,100,400,500,900,1000,4000};
        String s="";
        while(n!=0)
        {
            for(int i=0;i<A.length-1;i++)
            {
                if(A[i+1]>n)
                {
                    switch(A[i])
                    {
                        case 1:s+="I";break;
                        case 4:s+="IV";break;
                        case 5:s+="V";break;
                        case 6:s+="VI";break;
                        case 7:s+="VII";break;
                        case 8:s+="VIII";break;
                        case 9:s+="IX";break;
                        case 10:s+="X";break;
                        case 40:s+="XL";break;
                        case 50:s+="L";break;
                        case 90:s+="XC";break;
                        case 100:s+="C";break;
                        case 400:s+="CD";break;
                        case 500:s+="D";break;
                        case 900:s+="CM";break;
                        case 1000:s+="M";break;
                    }
                    n-=A[i];break;
                }
            }
        }
    }
}
```

```

}

    System.out.println("Equivalent roman number : "+ s);
}

int NTable(char s)
{
    switch(s)
    {
        case 'I':return 1;
        case 'V':return 5;
        case 'X':return 10;
        case 'L':return 50;
        case 'C':return 100;
        case 'D':return 500;
        case 'M':return 1000;
    }
    return o;
}

void toDecimal(String r)
{
    int sum=o;
    for(int i=0;i<r.length();i++)
    {
        if(i!=r.length()-1)
        {
            if(NTable(r.charAt(i))>=NTable(r.charAt(i+1)))
                sum+=NTable(r.charAt(i));
            else
                sum-=NTable(r.charAt(i));
        }
        else
            sum+=NTable(r.charAt(i));
    }
    System.out.println("Equivalent decimal number : "+sum);
}

public static void main()
{
    Convert ob = new Convert();
    Scanner sc=new Scanner(System.in);
}

```

```

System.out.println("Which number do you want to convert:Roman(1-3999) or Decimal?");
    char a=sc.next().charAt(0);
    System.out.println("Enter the number : ");
    switch(a)
    {
        case 'r':
        case 'R':String r=sc.next();
        ob.toDecimal(r);break;
        case 'd':
        case 'D':int n=sc.nextInt();
        ob.toRoman(n);break;
        default:System.out.println("Wrong input...");}
    }
}

```

**Variable Description Table12:**

Sl.no.	Variable	Datatype	Description
1	ob	Convert	Object of class to invoke functions.
2	sc	Scanner	Object of Scanner class to take user input
3	a	char	Stores the character to determine roman or decimal conversion
4	r	String	Stores Roman string
5	n	int	Stores decimal integer

## Output12:

```
BlueJ: Terminal Window - ISCPROJECT
Options
Which number do you want to convert:Roman(1-3999) or Decimal?
Roman
Enter the number :
MMMXLVIII
Equivalent decimal number : 3048
```

```
BlueJ: Terminal Window - ISCPROJECT
Options
Which number do you want to convert:Roman(1-3999) or Decimal?
Decimal
Enter the number :
999
Equivalent roman number : CMXCIX
```

## **QUESTION 13**

Write a program to convert a decimal number (whole number/fractional number) into equivalent Binary, Octal and Hexadecimal number form in a menu driven logic.

## **SOLUTION**

### **Algorithm 13:**

1. START
2. Takes the decimal user input
3. Takes the user's choice of conversion as input
4. Invokes functions depending on user's choice. Generates the binary or octal or hexadecimal number and prints
5. STOP

### **Program13:**

```
//PRG 13
import java.util.*;
class ConvertDecimal
{
    int convInt(int n,int base)
    {
        if(n==0)
            return 0;
        else
            return convInt(n/base,base)*10+n%base;
    }
    void convFrac(double n,int base)
    {

        int f;
        System.out.print(".");
        for(int i=0;i<10;i++)
        {
            f=(int)(n*base);
            System.out.print(f);
            n=n*base-f;
        }
    }
    void convIntHex(int n)
    {
        if(n==0)
        {}
        else
        {
            convIntHex(n/16);
            int r=n%16;
            switch(r)
            {
                case 15:System.out.print("F");break;
            }
        }
    }
}
```

```

case 14:System.out.print("E");break;
    case 13:System.out.print("D");break;
    case 12:System.out.print("C");break;
    case 11:System.out.print("B");break;
    case 10:System.out.print("A");break;
    default:System.out.print(r);
}
}
}

void convFracHex(double n)
{
    int f;
    System.out.print(".");
    for(int i=0; i<10;i++)
    {
        f=(int)(n*16);
        n=n*16-f;
        switch(f)
        {
            case 15:System.out.print("F");break;
            case 14:System.out.print("E");break;
            case 13:System.out.print("D");break;
            case 12:System.out.print("C");break;
            case 11:System.out.print("B");break;
            case 10:System.out.print("A");break;
            default:System.out.print(f);
        }
        if(n==0)break;
    }
}

public static void main()
{
    ConvertDecimal ob=new ConvertDecimal();
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the number : ");
    double n=sc.nextDouble();
    int a=(int)n;
    System.out.println("Convert to Binary/ Octal/ Hexadecimal??");
    char ch=sc.next().charAt(0);long l;
}

```

```

switch(ch)
{
    case 'O':
    case 'o':l=ob.convInt(a,8);
    System.out.print(l);ob.convFrac(n-a,8);break;
    case 'B':
    case 'b':l=ob.convInt(a,2);
    System.out.print(l);ob.convFrac(n-a,2);break;
    case 'H':
    case 'h':ob.convIntHex(a);ob.convFracHex(n-a);break;
    default :System.out.println("Wrong input");

}
}
}

```

**Variable Description Table13:**

Sl.no.	Variable	Datatype	Description
1	ob	ConvertDecimal	Object of class to invoke functions.
2	sc	Scanner	Object of Scanner class to take user input.
3	n	double	Stores the decimal number.
4	a	int	Stores the integer part of n
5	ch	char	Stores the character to determine binary or octal or hexadecimal conversion.
6	l	long	Stores the integer part of either binary or octal conversion

### Output13:

```
BlueJ: Terminal Window - ISCPROJECT
Options
Enter the number :
45.5
Convert to Binary/ Octal/ Hexadecimal??
Binary
101101.1000000000
```

```
BlueJ: Terminal Window - ISCPROJECT
Options
Enter the number :
67.76
Convert to Binary/ Octal/ Hexadecimal??
Octal
103.6050753412
```

```
BlueJ: Terminal Window - ISCPROJECT
Options
Enter the number :
269.8769
Convert to Binary/ Octal/ Hexadecimal??
Hexadeciml
10D.E07C84B5DC
```

### **QUESTION 14:**

Given a time in numbers we can convert it into words. For example

5 : 00 five o' clock

5 : 10 ten minutes past five

5 : 15 quarter past five

5 : 30 half past five

5 : 40 twenty minutes to six

5 : 45 quarter to six

5 : 47 thirteen minutes to six

Write a program which first inputs two integers, the first between 1 and 12 (both inclusive) and second between 0 and 59 (both inclusive) and then prints out the time they represent, in words. Your program should follow the format of the examples above.

### **SOLUTION:**

#### **Algorithm14:**

1. START
2. Takes the time as user input and stores in string s
3. Separates s into hrs and mins and stores in h and m respectively using StringTokenizer.
4. Uses the logic in printWords() function to print the time in words
5. STOP

## **Program14:**

```
//PRG 14
import java.util.*;
class TimeInWords
{
    static void printWords(int h, int m)
    {
        String nums[] = { "zero", "one", "two", "three", "four",
                          "five", "six", "seven", "eight", "nine",
                          "ten", "eleven", "twelve", "thirteen",
                          "fourteen", "fifteen", "sixteen", "seventeen",
                          "eighteen", "nineteen", "twenty", "twenty one",
                          "twenty two", "twenty three", "twenty four",
                          "twenty five", "twenty six", "twenty seven",
                          "twenty eight", "twenty nine" };

        if (m == 0)
            System.out.println(nums[h] + " o' clock ");

        else if (m == 1)
            System.out.println("one minute past " +
                               nums[h]);

        else if (m == 59)
            System.out.println("one minute to " +
                               nums[(h % 12) + 1]);

        else if (m == 15)
            System.out.println("quarter past " + nums[h]);

        else if (m == 30)
            System.out.println("half past " + nums[h]);
    }
}
```

```

else if (m == 45)
    System.out.println("quarter to " +
        nums[(h % 12) + 1]);

else if (m <= 30)
    System.out.println( nums[m] + " minutes past " +
        nums[h]);

else if (m > 30)
    System.out.println( nums[60 - m] + " minutes to " +
        nums[(h % 12) + 1]);
}

public static void main(String []args)
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the time = ");
    String s=sc.nextLine();
    StringTokenizer st=new StringTokenizer(s,":");
    int h = Integer.parseInt(st.nextToken().trim());
    int m = Integer.parseInt(st.nextToken().trim());
    printWords(h, m);
}
}

```

#### **Variable Description Table14:**

<b>Sl.no.</b>	<b>Variable</b>	<b>Datatype</b>	<b>Description</b>
1	sc	Scanner	Object of Scanner class to take user input
2	s	String	Stores the input time
3	st	StringTokenizer	Separates the input into hrs and mins using ':' as the delimiter
4	h	int	Stores the hour value
5	m	int	Stores the minute value

## Output14:

BlueJ: Terminal Window - ISCPROJECT

Options

Enter the time =  
5:45  
quarter to six

BlueJ: Terminal Window - ISCPROJECT

Options

Enter the time =  
6:01  
one minute past six

BlueJ: Terminal Window - ISCPROJECT

Options

Enter the time =  
12 : 30  
half past twelve

BlueJ: Terminal Window - ISCPROJECT

Options

Enter the time =  
4:37  
twenty three minutes to five

### **QUESTION 15:**

A class Personal contains employee details and another class Retire calculates the employee's Provident Fund and Gratuity. The details of the two classes are given below:

Class name : Personal

Instance variables:

Name : stores the employee name

Pan : stores the employee PAN

basic\_pay : stores the employee basic salary ( in decimals )

acc\_no : stores the employee bank account number

Member methods:

Personal(....) : parameterized constructor to assign value to data members

void display( ) : to display the employee details

Class name : Retire

Instance variables:

yrs : stores the employee years of service

pf : stores the employee provident fund amount (in decimals)

grat : stores the employee gratuity amount (in decimals)

Member methods:

Retire ( ..... ) : parameterized constructor to assign value to data members of both the classes.

void provident( ) : calculates the PF as (2% of the basic pay) \* years of service.

void gratuity( ) : calculates the gratuity as 12 months salary, if the years of service is > 10 years else the gratuity amount is nil.

`void display ()` : Display the employee details along with the Provident Fund and gratuity amount.

Specify the class Personal giving details of the constructor and member functions `void display()`.

Using the concept of inheritance, specify the class Retire giving details of constructor, and the member functions `void provident()`, `void gratuity()` and the `void display()`.

### **SOLUTION:**

#### **Algorithm15:**

1. START
2. Takes the name, Pan number, basic pay, account number and number of years of service of the employee as input in the `main()` method of Retire class, the child class of Personal class.
3. Calculates the provident fund and gratuity of the employee and calls the `display()` function which displays all the user info and the calculated amounts.
4. STOP

## **Program15:**

```
//PRG 15
import java.util.*;
class Personal
{
    String Name,Pan;double basic_pay;long acc_no;
    Personal(String N,String P,double bp,long an)
    {
        Name=N;Pan=P;basic_pay=bp;acc_no=an;
    }
    void display()
    {
        System.out.println("Personal Details :");
        System.out.println("Name : "+Name+"\nPan number :
"+Pan+"\nBasic Pay : "+basic_pay+"\nAccount number : "+acc_no);

    }
}
import java.util.*;
class Retire extends Personal
{
    int yrs;double pf,grat;
    Retire(String N,String P,double bp,long an,int y)
    {
        super(N,P,bp,an);
        yrs=y;
    }
    void provident()
    {
        pf=0.02*super.basic_pay*yrs;
    }
    void gratuity()
    {
        if(yrs>10)
            grat=12*super.basic_pay;
        else
            grat=0;
    }
}
```

```
void display()
{
    super.display();
    System.out.println("Provident Fund : "+pf+"\nGratuity : "+grat);
}
public static void main()
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter your name, Pan number, basic pay,
account number, years of service :");
    String N=sc.nextLine();
    String P=sc.next();
    double bp=sc.nextDouble();
    long an=sc.nextLong();
    int y=sc.nextInt();
    Retire ob=new Retire(N,P,bp,an,y);
    ob.provident();
    ob.gratuitiy();
    ob.display();
}
}
```

### **Variable Description Table15:**

<b>Sl.no.</b>	<b>Variable</b>	<b>Datatype</b>	<b>Description</b>
1	Name	String	Stores the name of
2	Pan	String	stores the employee PAN
3	basic_pay	double	stores the employee basic salary ( in decimals )
4	acc_no	long	stores the employee bank account number
5	yrs	int	stores the employee years of service
6	pf	double	stores the employee provident fund amount (in decimals)
7	grat	double	stores the employee gratuity amount (in decimals)

### **Output15:**

```
BlueJ: Terminal Window - ISCPROJECT
Options
Enter your name, Pan number, basic pay, account number, years of service :
Subhranil Nandy
Y242Y3FYFTT4UFU50H3B587
50000
6595656526956668765
15
Personal Details :
Name : Subhranil Nandy
Pan number : Y242Y3FYFTT4UFU50H3B587
Basic Pay : 50000.0
Account number : 6595656526956668765
Provident Fund : 15000.0
Gratuity : 600000.0
```

### **QUESTION 16.**

Design three classes: ClassTest, FinalExam and Result. The ClassTest class has data members representing marks of four-unit tests and member function to accept and display the data. The class FinalExam will inherit the class ClassTest has data member to hold the marks of Final Exam and member function to accept and display the data. Derive the Result class from the class FinalExam and add a data member to hold the marks of final exam. The class Result with a single data member total\_marks and a member function to compute the total by adding 20% of total of the fourunit tests and 80% of the final exam marks and store it in its data member total\_marks. It has also a function to display the total marks. In the main method, create an object of Result class and call the functions accordingly.

### **SOLUTION:**

#### **Algorithm: -**

\* Step 1:-Start.

\* Step 2:-

In ClassTest create a constructor to initialize marks in four Class Test.

\* Step 3:-Create a diplay function to print them.

\* Step 4:-

In FinalExam create a constructor to initialize marks in Final Exam and pass rest values to super class.

\* Step 5:-Create a diplay function to print it and call display of ClassTest.

\* Step 6:-In Result create a function to compute the total marks and save it.

\* Step 7:-

Create a diplay function to print it and call the display function in FinalExam.

\* Step 8:-

Create a main menu to input all data from user and pass the values to constructor of super class.

\* Step 9:-Now call the display function.

\* Step 10:-Stop.

### Program 16:

```
//PRG16
import java.util.*;
class ClassTest
{
    double c1,c2,c3,c4;//To store marks in Class Tests.
    ClassTest(double a,double b,double c,double d)
    {
        c1=a;//Initializing Variables.
        c2=b;
        c3=c;
        c4=d;
    }//end of constructor.
    void display()
    {
        //Displaying the marks in Class Test.
        System.out.println("The marks in Class Test: "+c1+","+c2+","+c3+","+c4);
    }//end of display.
}//end of class.

import java.util.*;
class FinalExam extends ClassTest
{
    double f;//To store marks in Final Exams.
    FinalExam(double a,double b,double c,double d,double e)
    {
        super(a,b,c,d);//Initializing Variables.
        f=e;
    }//end of constructor.
    void display()
    {
        //Displaying the marks in Class Test and Final Exam.
        super.display();
        System.out.println("The marks in Final Exam is : "+f);
    }//end of display.
}//end of class.

import java.util.*;
class Result extends FinalExam
{
    double total_marks;//To store total marks.
    Result(double a,double b,double c,double d,double e)
    {
```

```
//end of constructor.  
void compute()  
{  
    //Calculating total marks.  
    total_marks=Math.round((0.2*(c1+c2+c3+c4))+(0.8*f));  
}//end of compute.  
void display()  
{  
    //Displaying the marks in Class Test and Final Exam and total marks.  
    super.display();  
    System.out.println("The total marks is: "+total_marks);  
}//end of display.  
public static void main(String args[])  
{  
    Scanner sc=new Scanner(System.in);  
    System.out.println("Enter the marks in four Class Tests");  
    double a=sc.nextDouble();//To store marks in 1st Class Test.  
    double b=sc.nextDouble();//To store marks in 2nd Class Test.  
    double c=sc.nextDouble();//To store marks in 3rd Class Test.  
    double d=sc.nextDouble();//To store marks in 4th Class Test.  
    System.out.println("Enter the marks in Final Exams");  
    double e=sc.nextDouble();//To store marks in Final Exams.  
    Result R=new Result(a,b,c,d,e);//Creating class object.  
    R.compute();//Calling required functions.  
    R.display();  
}//end of main.  
}//end of class.
```

## Variable Description Chart:

Sl.no	Variable	Datatype	Usage
1	sc	Scanner	Object of Scanner class
2	cl1	double	To store marks in class test 1
3	cl2	double	To store marks in class test 2
4	cl3	double	To store marks in class test 3
5	cl4	double	To store marks in class test 4
6	f	double	To store total marks in final exams
7	total_marks	double	To store total marks
8	a	double	To store marks in class test 1
9	b	double	To store marks in class test 2
10	c	double	To store marks in class test 3
11	d	double	To store marks in class test 4
12	e	double	To store total marks in final exams
13	R	Result	Object of class to invoke functions accordingly

### Output:

BlueJ Terminal Window - Class12

Options

Enter the marks in four Class Tests

45 49 43 50

Enter the marks in Final Exams

95

The marks in Class Test: 45.0,49.0,43.0,50.0

The marks in Final Exam is : 95.0

The total marks is: 113.0

### QUESTION 17.

In a computer game, a vertical column and a pile of rings are displayed. The objective of the game is to pile up rings on the column till it is full. It can hold 10 rings at the most. Once the column is full, the rings have to be removed from the top till the column is empty and then the game is over. Define the class RingGame with the following details: Class name : RingGame

Data members/instance variables

ring [ ] : array to hold rings (integer) MAX : integer to hold maximum capacity of ring array

TOP : integer to point to the upper most element

Member functions :- RingGame(int m) constructor to initialize, MAX = m & TOP to -1.

void jump-in(int ) adds a ring to the top of the column, if possible, otherwise displays a message "Column full. Start removing rings".

void jump-out( ) removes the ring from the top, if column is not empty otherwise, outputs a message, "Congratulations. The game is Over". Specify the class RingGame giving the details of the constructor and functions

void jumpin(int) and void jump-out( ). Also define the main function to create an object and call methods accordingly to enable the task.

### SOLUTION:

**Algorithm:** -

\* Step 1:-Start.

\* Step 2:-

Enter the size of ring from user and declare a array for storing rings with it.

\* Step 3:-

Create a function jumpin and perform the following operation to insert elements: If upper is one less than max print "Column full ! Start removing rings.", else insert the element passed and print the elements of array.

\* Step 4:-

Create a function jumpout and perform the following operation to remove elements: If upper is equal to -

1 print "Congratulations ! the game is over .", else remove the element last inserted and print the elements of array.

\* Step 6:-Take inputs from user and execute them till game is over.

\* Step 7:-Stop.

### Program 17:

```
//PRG17
import java.util.*;
class RingGame
{
    int max;//Initializing data members.
    int[] ring;
    int upper;
    RingGame(int m)
    {
        max=m;//Initializing data values.
        ring=new int[m];
        upper=-1;
    }//end of Constructor.
    void jumpin(int i)
    {
        if(upper==max-1)//ring will not be inserted.
        {
            System.out.println("Column full ! Start removing rings.");
        }//end of if.
        else
        {
            ring[++upper]=i;//Preincrementation used so that the first element is
            inserted at index 0.

            for(int x=0;x<=upper;x++)//Printing Ring.
            {
                System.out.print(ring[x]+" ");
            }//end of for.
            System.out.println();
        }//end of else.
    }//end of jumpin.
    void jumpout()
    {
        if(upper==-1)//game is over.
        {
            System.out.println("Congratulations ! the game is over .");
            System.exit(0);
        }
    }
}
```

```
//end of if.  
else  
{  
    ring[upper]=ring[upper--];//element last inserted is popped out.  
    for(int x=0;x<=upper;x++)//Printing Ring.  
    {  
        System.out.print(ring[x]+" ");  
    }//end of for.  
    System.out.println();  
}//end of else.  
}//end of jumpout.  
public static void main (String[] args)  
{  
    RingGame obj = new RingGame(3);  
    Scanner sc=new Scanner(System.in);  
    System.out.println("Enter 1 for adding a ring");  
    System.out.println("Enter 2 for removing a ring");  
    for(int i=1;i>0;i++)  
    {  
        System.out.println("Enter your choice");  
        int c=sc.nextInt();  
        if(c==1)  
        {  
            System.out.println("Enter element to be added in Ring");  
            int n=sc.nextInt();  
            obj.jumpin(n);//To add rings.  
        }  
        else if(c==2)  
        {  
            obj.jumpout();//To remove rings.  
        }  
    }  
}//end of main.  
}//end of class.
```

### Variable Description Chart 17:

Sl.no	Variable	Datatype	Usage
1	sc	Scanner	Object of Scanner class
2	max	int	To store array size
3	ring	int[]	To store ring elements
4	upper	int	To store upper limit
5	m	int	To initialize array size
6	x	int	To run element display loop
7	c	int	To store choice of user
8	n	int	To store element to be inserted
9	obj	RingGame	Object of class to invoke functions accordingly

### Output 17:

```
BlueJ Terminal Window - Class12
Options
Enter 1 for adding a ring
Enter 2 for removing a ring
Enter your choice
1
Enter element to be added in Ring
34
34
Enter your choice
1
Enter element to be added in Ring
89
34 89
Enter your choice
1
Enter element to be added in Ring
67
34 89 67
Enter your choice
2
34
Enter your choice
2

Enter your choice
2
Congratulations ! the game is over .
```

### QUESTION 18:

Declare a class named CircularQueue with following declaration:-

Class Name - CircularQueue

Data members:- double Q[ ] - an array to hold floating points.

int N - the size of the array (max size 10)

int front, rear - two pointers of the queue

Member methods:-

CircularQueue (int n) - constructor to initialize size = n and front & rear with 0 and the array elements with default value.

void enqueue(double nm ) - to push one number to the array, if the queue is full display the message "OVERFLOW"

double dequeue( ) - to extract one number from the array. If the queue is empty, display the message "UNDERFLOW"

void display( ) - to display content of the array. Specify the above class with all the member methods. Write the main function to create an object and call the functions accordingly.

### SOLUTION:

#### Algorithm18: -

\* Step 1:-Start.

\* Step 2:-

Enter user's choice to remove or insert an element, also take the element from user.

\* Step 3:-

Create a function enqueue and perform the following operation to insert elements:

Step A:-If(rear+1)%max is equal to front, print "Overflow" and go to step D.

Step B:-If front is equal to -1 and rear is equal to -1, set both as 0.

Else make rear=max-1

and front not equal to 0, then set rear as 0. Else rear=rear+1%max.

Step C: Set value of queue at rear as val.

Step D: Exit.

\* Step 4:-

Create a function dequeue and perform the following operation to remove elements:

**Step A:** If front is -1. Print "Underflow" and go to Step D.  
**Step B:** Set val as queue at front.  
**Step C:** IF front and rear are equal set front and rear as -1.  
Else if front is one less than max set front as 0. Else increment front by 1.  
**Step D:** Exit  
**\*Step 5:-Stop.**

Program  
-----  
#include <iostream.h>  
#include <conio.h>  
#define max 10  
int arr[max];  
int front = -1, rear = -1;  
void enque(int);  
void deque();  
void display();  
void main()  
{  
 int choice, val;  
 clrscr();  
 do  
 {  
 cout << "1.Enqueue" << endl;  
 cout << "2.Dequeue" << endl;  
 cout << "3.Display" << endl;  
 cout << "4.Exit" << endl;  
 cout << "Enter your choice : ";  
 cin >> choice;  
 switch(choice)  
 {  
 case 1:  
 enque();  
 break;  
 case 2:  
 deque();  
 break;  
 case 3:  
 display();  
 break;  
 case 4:  
 exit(0);  
 default:  
 cout << "Wrong choice" << endl;  
 }  
 } while(choice != 4);  
}

### Program 18:

```
//PRG18
import java.util.*;
class CQueue
{
    double queue[];//array declaration.
    int N;//To store Array size
    int front=0;
    int rear=0;
    CQueue(int n)
    {
        front=-1;//Initializing data members.
        rear=-1;
        N=n;
        queue=new double[N];
    }//end of constructor.
    void enqueue(double nm)//function to insert an element in a circular
queue.
    {
        if(front== -1 && rear== -1)//condition to check queue is empty.
        {
            front=0;
            rear=0;
            queue[rear]=nm;
        }//end of if.
        else if((rear+1)%N==front)//condition to check queue is full.
        System.out.println("Queue is overflow.");
        else
        {
            rear=(rear+1)%N;//rear is incremented.
            queue[rear]=nm;//assigning a value to the queue at the rear
position.
        }//end of else.
    }//end of enqueue.
    double dequeue()//function to delete the element from the queue.
    {
        if((front== -1) && (rear== -1))//condition to check queue is empty.
        System.out.println("Queue is underflow.");
        else if(front==rear)
```

```

System.out.println( Queue is underflow. );
else if(front==rear)
{
    System.out.println("The dequeued element is "+ queue[front]);
    front=-1;
    rear=-1;
}//end of else if.
else
{
    System.out.println("The dequeued element is "+ queue[front]);
    front=(front+1)%N;
}//end of else.
return o;//returning value.
}//end of dequeue.
void display()//function to display the elements of a queue.
{
    int i=front;
    if(front===-1 && rear===-1)
    {
        System.out.println("Queue is empty..");
    }//end of if.
    else
    {
        System.out.println("Elements in a Queue are :");
        while(i<=rear)
        {
            System.out.println( queue[i]);
            i=(i+1)%N;
        }//end of while loop.
    }//end of else.
}//end of display.
public static void main()
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter maximum value:");
    int n=sc.nextInt();//to store array size.
    CQueue C=new CQueue(n);
    System.out.println("Press 1:Insert an element");
    System.out.println("Press 2:Delete an element");
    System.out.println("Press 3:Display the element");
    System.out.println("Press 4>To end program");
    int c=1;
    double x=0;//variables declaration.
    while(c<4 && c!=0)//while loop.
    {
}

```

```

i
System.out.println("Enter your choice");
c=sc.nextInt(); //To store the choice of user.
switch(c)
{
    case 1:
        System.out.println("Enter the element which is to be inserted");
        x=sc.nextDouble(); //Storing element to be inserted.
        C.enqueue(x); //Calling required functions.
        break;
    case 2:
        C.dequeue();
        break;
    case 3:
        C.display();
    } //end of switch case.
} //end of while loop.
} //end of main.
} //end of class.

```

**Variable Description Table18:**

Sl.no	Variable	Datatype	Usage
1	sc	Scanner	Object of Scanner class
2	queue	double[]	To hold circular queue
3	N	int	To initialize size of queue
4	front	int	To store rear element position
5	rear	int	To store front element position
6	nm	double	To store element to be inserted
7	i	int	To store front position
8	n	int	To take array size as input from user
9	c	int	To store user's choice
10	x	double	To take element as input from user for insertion

u	C	CQueue	Object of class to invoke functions accordingly
---	---	--------	---

Output 18:

```

Enter maximum value:
3
Press 1:Insert an element
Press 2>Delete an element
Press 3:Display the element
Press 4>To end program
Enter your choice
1
Enter the element which is to be inserted
34
Enter your choice
1
Enter the element which is to be inserted
54
Enter your choice
1
Enter the element which is to be inserted
76
Enter your choice
1
Enter the element which is to be inserted
90
Queue is overflow.
Enter your choice
2
The dequeued element is 34.0
Enter your choice
2
The dequeued element is 54.0
Enter your choice
2
The dequeued element is 76.0
Enter your choice
2
Queue is underflow.
Enter your choice
4

```

### **QUESTION 19:**

Write a program to simulate conversion of numbers from Decimal number system to Binary/Octal/Hexadecimal equivalent and vice-versa as per the user's choice. The program should display the menu accordingly and ask the user for proper input. The program should have proper exception handling mechanism and validation checking.

### **SOLUTION:**

#### **Algorithm19:**

- \* Step 1:-Start.
- \* Step 2:-Take the user choice and number as input.
- \* Step 3:-Create the following functions to perform the asked operation.
  - \* For Decimal to Binary keep dividing the number by 2 until 0 or 1 is left. Store the remainder in every iteration and store in reverse order.
  - \* For Decimal to Octal keep dividing the number by 8 until a number between 0 to 7 is left. Store the remainder in every iteration and store in reverse order.
  - \* For Decimal to Hexadecimal keep dividing the number by 16 until a number between 0 to 15 is left. Store the remainder in every iteration and store in reverse order. Store number as A to F for 10 to 15.
  - \* For Decimal to binary keep dividing the number by 2 until 0 or 1 is left. Store the remainder in every iteration and store in reverse order.
- \* For converting from any system to Decimal use an operation `Integer.parseInt(N,x)` where N is string containing the number input and x is 2 for binary,8 for octal and 16 for hexadecimal input. Store it in a variable.
- \* Step 4:-Display the output.
- \* Step 5:-Stop.

### Program 19:

```
//PRG19
import java.util.*;
class Conversion
{
    void Bin(int N)
    {
        int bin[] = new int[100]; //array to store binary number.
        int i=0; //counter for binary number array.
        while(N>0)
        {
            bin[i++]=N%2; //storing remainder in binary array.
            N=N/2;
        }
        System.out.print("Binary number is : ");
        for(int j=i-1;j>=0;j--)
            System.out.print(bin[j]); //Printing binary number array in reverse order.
    } //end of Bin.
    void Oct(int N)
    {
        int[] octalNum = new int[100]; //array to store octal number.
        int i=0; //counter for octal number array.
        while(N!=0)
        {
            octalNum[i]=N % 8; //storing remainder in octal array.
            N=N/8;
            i++;
        }
        for (int j = i-1; j >= 0; j--)
            System.out.print(octalNum[j]); //Printing octal number array in reverse
        order.
    } //end of Oct.
    void Hex(int N)
    {
        char[] hexaDeciNum = new char[100]; //char array to store hexadecimal
        number.
```

```

{
    int temp = 0; //temporary variable to store remainder.
    temp=N%16; //storing remainder in temp variable.
    if (temp<10)//check if temp < 10.
    {
        hexaDeciNum[i]=(char)(temp+48);
        i++;
    }
    else
    {
        hexaDeciNum[i]=(char)(temp+55);
        i++;
    }
    N=N/16;
}
for (int j=i-1;j>=0;j--)
    System.out.print(hexaDeciNum[j]); //printing hexadecimal number array
in reverse order.
}//end of Hex.
void Dec(String N,int p) //Converting to decimal Number.
{
    int dec=0; //to store decimal number.
    if(p==1)
    {
        dec=Integer.parseInt(N,2);
        System.out.println("Decimal Value is: "+dec);
    }
    if(p==2)
    {
        dec=Integer.parseInt(N,8);
        System.out.println("Decimal Value is: "+dec);
    }
    if(p==3)
    {
        dec=Integer.parseInt(N,16);
        System.out.println("Decimal Value is: "+dec);
    }
}//end of Dec.
public static void main(String args[])
{
    Conversion N=new Conversion();

```

```
dec=Integer.parseInt(N,16);
System.out.println("Decimal Value is: "+dec);
Conversion N=new Conversion();
Scanner sc=new Scanner(System.in);
```

```
System.out.println("Enter 3 for Octal input");
System.out.println("Enter 4 for Hexadecimal input");
int c1=sc.nextInt(); //To store type of input.
for(int i=0;c1==2;i++)
{
    System.out.println("Enter the Number: ");
    int n=sc.nextInt(); //To store the number.
    System.out.println("Enter 1 for Binary output");
    System.out.println("Enter 2 for Octal output");
    System.out.println("Enter 3 for Hexadecimal output");
    System.out.println("Enter 0 to end program");
    int dc=sc.nextInt(); //To store user choice.
    if(dc==1)//Calling necessary functions.
        N.Bin(n);
    else if(dc==2)
        N.Oct(n);
    else if(dc==3)
        N.Hex(n);
    else if(dc==0)
        System.exit(0);
    System.out.println(); //To shift line after each output.
}//end of for loop.
System.out.println("Enter the number");
String n=sc.next(); //To store the input.
if(c1==1)//Calling necessary functions.
    N.Dec(n,1);
else if(c1==3)
    N.Dec(n,2);
else if(c1==4)
    N.Dec(n,3);
else
    System.out.println("Invalid Input");
}//end of main.
}//end of class.
```

**Variable Description Table19:**

Sl.no	Variable	Datatype	Usage
1	sc	Scanner	Object of Scanner class
2	N	int	To store decimal number to be converted
3	Bin	int[]	To store Binary number
4	Octal	int[]	To store Octal number
5	hexaDeciNum	char[]	To store Hexadecimal number
6	i	int	To act as an array counter
7	j	int	To run a loop for printing number
8	temp	int	Temporary variable which stores remainder
9	N	String	To store Binary ,octal or hexadecimal number
10	dec	int	To store decimal number
11	p	int	To store condition of input number
12	c1	int	To store type of input
13	n	int	To store number entered
14	dc	int	To store user's choice
15	N	Conversion	Object of class to invoke functions accordingly

### Output 19:

```
BlueJ: Terminal Window - Class12
Options
Enter 1 for Binary input
Enter 2 for Decimal input
Enter 3 for Octal input
Enter 4 for Hexadecimal input
2
Enter the Number:
47
Enter 1 for Binary output
Enter 2 for Octal output
Enter 3 for Hexadecimal output
Enter 0 to end program
1
Binary number is : 101111
Enter the Number:
47
Enter 1 for Binary output
Enter 2 for Octal output
Enter 3 for Hexadecimal output
Enter 0 to end program
2
57
Enter the Number:
47
Enter 1 for Binary output
Enter 2 for Octal output
Enter 3 for Hexadecimal output
Enter 0 to end program
3
2F
Enter the Number:
47
Enter 1 for Binary output
Enter 2 for Octal output
Enter 3 for Hexadecimal output
Enter 0 to end program
0
```

BlueJ: Terminal Window - Class12

Options

Enter 1 for Binary input

Enter 2 for Decimal input

Enter 3 for Octal input

Enter 4 for Hexadecimal input

1

Enter the number

1010

Decimal Value is: 10

BlueJ: Terminal Window - Class12

Options

Enter 1 for Binary input

Enter 2 for Decimal input

Enter 3 for Octal input

Enter 4 for Hexadecimal input

4

Enter the number

AA2FF

Decimal Value is: 697087

Java.org

## **QUESTION 20.**

Write a program to simulate encryption and decryption of a given string. The program should be capable of converting a given string to a coded message and also be capable of decoding the same if given again. [Students can use any formula of their choice for encryption and decryption of the string]. The program should display the menu accordingly and ask the user for proper input.

## **SOLUTION:**

### **Algorithm 20: -**

- \* Step 1:-Start.
- \* Step 2:-Enter string input from user to be encoded.
- \* Step 3:Create a function Encryption and perform the following operation:
  - ◊Find the length L of the string.
  - ◊Find the ceil and floor values of root Length and assign them to the variables.
  - ◊Check if the product of the two variables  $\geq$  Length, if not then increments the variable having a lesser value by 1.
  - ◊Create a 2D matrix and fill the characters of the string row-wise.
  - ◊Read the matrix columnwise to get the encrypted string.
- \* Step 4:-Enter string input from user to be decoded.
- \* Step 5:Create a function Decryption and perform the following operation:
  - ◊Find the length L of the string.
  - ◊Find the ceil and floor values of root Length and assign them to the variables.
  - ◊Create a 2D matrix and fill the matrix by characters of string column-wise.
  - ◊Read the matrix row-wise to get the decrypted string.
- \* Step 6:-Stop.

### **Program 20:**

```
//PRG 20
import java.util.*;
class CaesarCipher
{
    String st;char ch;
    CaesarCipher()
    {
        st="";
    }
    void input()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the sentence : ");
        st=sc.nextLine();
        System.out.println("Do you want to encode or decode the string?");
        ch=sc.next().charAt(0);
    }
    String encode()
    {
        String enc="";
        for(int i=0;i<st.length();i++)
        {
            char z=st.charAt(i);
            if(z==' ' || z=='.')
            {
                enc=enc+z;
                continue;
            }

            z++;
            enc=enc+z;
        }
        return enc;
    }
    String decode()
    {
```

```

String dec="";
for(int i=0;i<st.length();i++)
{
    char z=st.charAt(i);
    if(z==' ' || z=='.')
    {
        dec=dec+z;
        continue;
    }
    z--;
    dec=dec+z;
}
return dec;
}
public static void main()
{
    CaesarCipher ob=new CaesarCipher();
    ob.input();
    if(ob.ch=='E' || ob.ch=='e')
        System.out.println("Encoded word = " + ob.encode());
    else if(ob.ch=='D' || ob.ch=='d')
        System.out.println("Decoded word = " + ob.decode());
    else
    {
        System.out.println("Wrong input...");
        System.exit(0);
    }
}
}

```

### Variable Description Table20:

Sl.no	Variable	Datatype	Usage
1	sc	Scanner	Object of Scanner class
2	st	String	To store user inputted string
3	ch	char	To store user's choice
4	enc	String	To store encoded string
5	i	int	Loop control variable
6	z	char	To store each letter of st
7	dec	String	To store decoded string
8	ob	CaesarCipher	Object of class to invoke functions accordingly

**Output20:**

④ BlueJ: Terminal Window - Class12

---

Options

---

Enter the sentence :

Quick brown fox.

Do you want to encode or decode the string?

Encode

Encoded word = Rvjd1 cspxo gpy.

④ BlueJ: Terminal Window - Class12

---

Options

---

Enter the sentence :

Tvcisbojm Oboez.

Do you want to encode or decode the string?

Decode

Decoded word = Subhranil Nandy.

## BIBLIOGRAPHY

- **Objects First with Java (- David J. Barnes & Michael Kölbing)**
- **Computer Science with JAVA (-Sumita Arora)**
- **Guided Computer Science (-S.K. Gaur & M. Vijay Bhaskar)**