TABLE OF CONTENTS

Ex:No: 1 **GOLDBACH NUMBER**
Date:

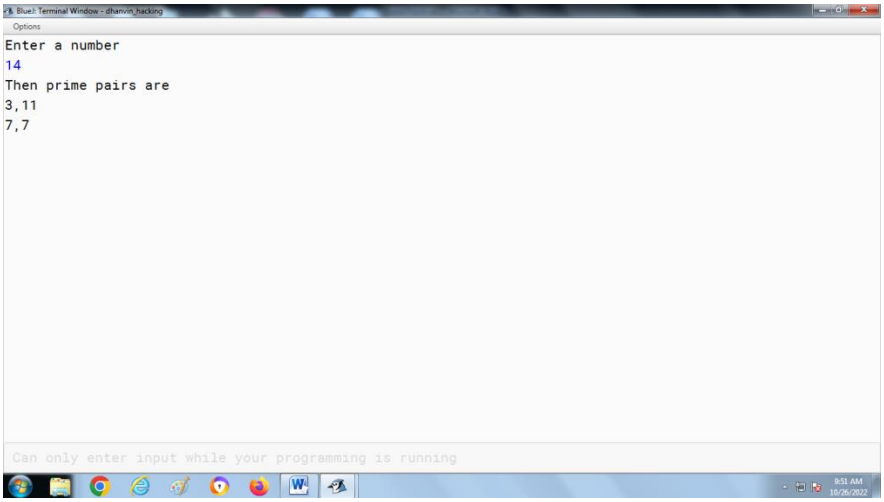
Aim:

To find all the possible sum of odd pairs, whose answer is equal to the given even number.

Source Code:

```
import java.util.*;
public class Goldbachnumber{
public static boolean isPrime(int num){
int c=0;
for(int i=1;i<=num;i++){
if(num%i==0){
c++;
}
}
return c == 2;
}
public static void main(String args[]){
Scanner sc=new Scanner (System.in);
System.out.println("Enter a number");
int n=sc.nextInt();
if(n<=9 || n>=50){
System.out.println("Invalid input: Number out of range");
}
if(n % 2 !=0){
System.out.println("Invalid input: Number is odd");
}
System.out.println("Then prime pairs are");
int a=3;
int b=0;
while(a<=n/2){
b= n - a;
if(isPrime(a) && isPrime(b)){
System.out.println(a + "," + b);
}
a+=2;
}
}
}
```

Output:



Conclusion:

Thus we have found all the possible sum of odd pairs, whose answer is equal to the given even number.

Ex No: 2 **SUM OF DIGITS**
Date:

Aim: -

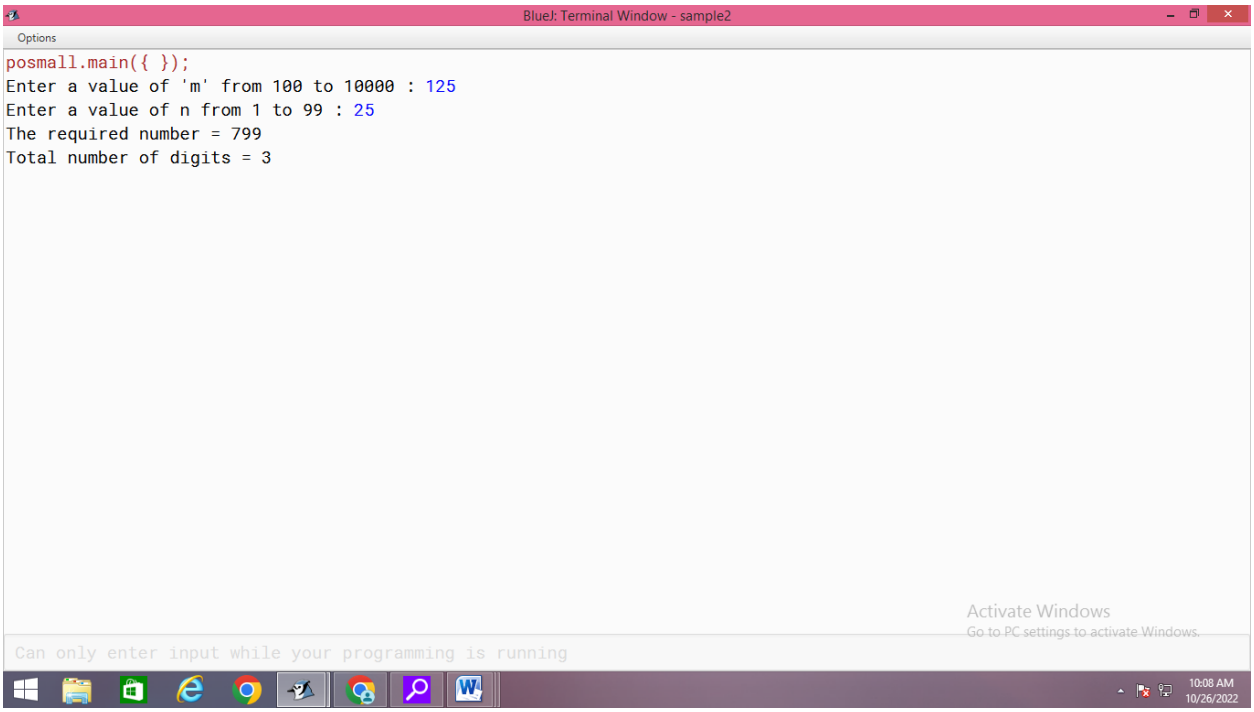
To accept numbers M and N from the user and print the smallest required number whose sum of all its digits is N.

Source Code:

```
import java.util.*;
public class posmall{
int sumDig(int n)
   {
      int sum = 0, d;
      while(n>0)
      {
         d = n%10;
         sum = sum + d;
         n = n/10;
      }
      return sum;
   }
    int countDig(int n)
   {
      String s = Integer.toString(n);
      int len = s.length();
      return len;
   }
    public static void main(String args[])
   {
      posmall ob = new posmall();
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter a value of 'm' from 100 to 10000 : ");
      int m = sc.nextInt();
      System.out.print("Enter a value of n from 1 to 99 : ");
      int n = sc.nextInt();

      if(m<100 || m>10000 || n<1 || n>99)
      {
         System.out.println("Invalid Input");
      }
      else
      {
         int i = m;
         while(ob.sumDig(i)!=n)
         {
```

```
            i=i+1;
        }
        System.out.println("The required number = "+i);
        System.out.println("Total number of digits = "+ob.countDig(i));
    }
  }
}
```

Output:



Conclusion:

Thus, the required smallest number is found.

Ex No.3 **Rotation Matrix**
Date:

Aim:

To find the rotated matrix and calculate the sum of corner elements

Source Code:

```
import java.util.*;
public class RotateMatrix
{
    public static void main (String args[])
    {
        int M=0;
        Scanner sc = new Scanner(System.in);
        if(M>3||M<9)
        {
        System.out.println("Enter the size of the matrix");
        }
        M=sc.nextInt();
        int arr[][]= new int[M][M];
        System.out.println("Enter the matrix elements");
        for(int i=0;i<M;i++)
        {
            for(int j=0;j<M;j++)
            {
                arr[i][j]=sc.nextInt();
            }
        }

        for(int i=0;i<M;i++)
        {
            for(int j=0;j<M;j++)
            {
                System.out.print(arr[i][j]+"\t");
            }
            System.out.println();
        }
        System.out.println("The rotated matrix is:");
        for(int i=0;i<M;i++)
        {
            for(int j=M-1;j>=0;j--)
            {
                System.out.print(arr[j][i]+"\t");
            }
```

```
            System.out.println();
        }
          {
            int sum = arr[0][0] + arr[0][M-1] + arr[M-1][0] + arr[M-1][M-1];
              System.out.println("Sum of the corner elements="+sum);

            }
}
```

Output:

Conclusion:

Thus the matrixrotation and the sum of corner elementshave been executed with the help of the above code.

Ex:No: 4 **Caesar Cipher**
Date:

Aim:

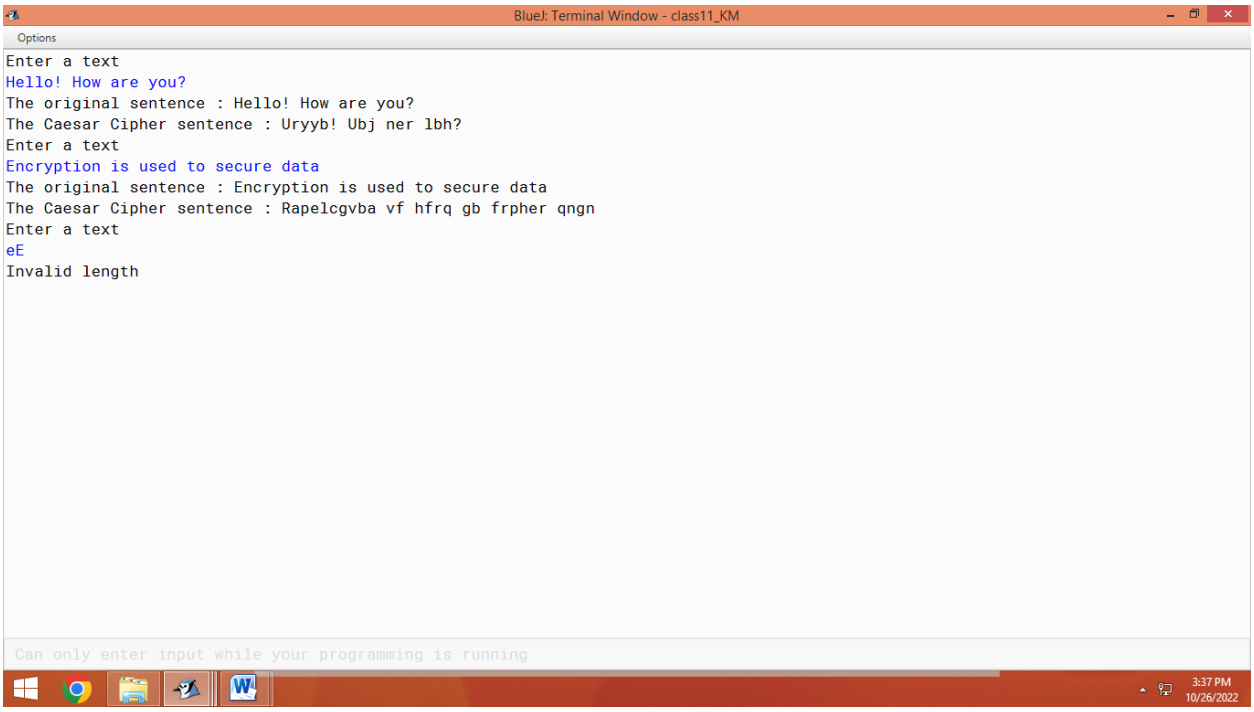
To convert the given text to its Caesar Cipher equivalent.

Source code:

```
import java.util.*;
public class Caesar{
   int L;
   String text;
   String convert;
   Caesar(){
      L = 0;
      text = "";
      convert = "";
   }
   void input(){
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter a text");
      text = sc.nextLine();
      L = text.length();
      if(L<=3||L>=100)
      {
         System.out.println("Invalid length");
         System.exit(0);
      }   }
   void conversion(){
      int temp;
      for(int i=0;i<L;i++)
      {
         if(text.charAt(i)>=65 && text.charAt(i)<=77 || text.charAt(i)>=97 &&
text.charAt(i)<=109)
         {
            temp = (int)text.charAt(i) + 13;
            char a = (char)temp;
            convert = convert + a;
         }else if(text.charAt(i)>=78 && text.charAt(i)<=90 || text.charAt(i)>=110 &&
text.charAt(i)<=122)
         {
            temp = (int)text.charAt(i) - 13;
            char a = (char)temp;
            convert = convert + a;
         }else{
            convert = convert + text.charAt(i);
```

```
            }
        }
    }
    void display(){
        System.out.println("The original sentence : " + text);
        System.out.println("The Caesar Cipher sentence : " + convert);
    }

    public static void main(String args[]){
        Caesar ob = new Caesar();
        ob.input();
        ob.conversion();
        ob.display();
    }
}
```

Output:



Conclusion:

Thus the given text was converted to its Caesar Cipher equivalent

Ex.No: 5 **Sorting the String**
Date:

Aim ::

To Arrange the given Sentence in Ascending Order of their Length.

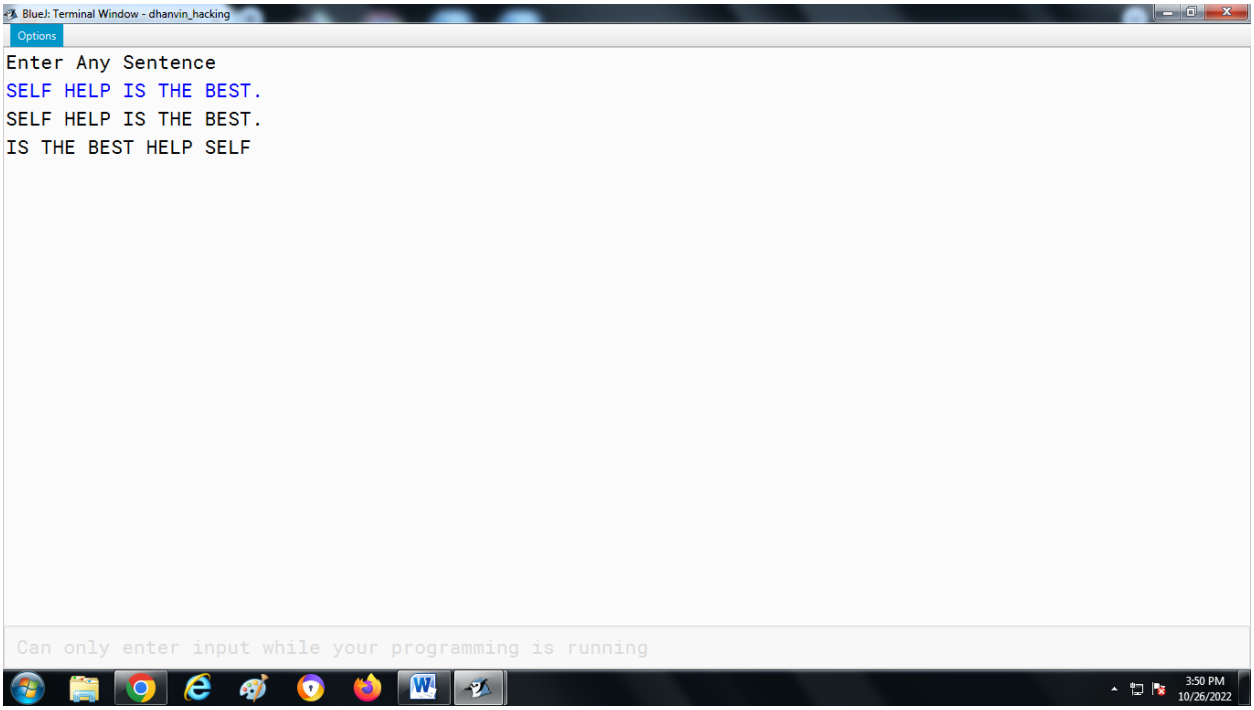
Source Code:

```
import java.util.*;
class Stringp
{
public static void main(String rags[])
{
    String x="",x1="";
    int s=0,p=0,h=0,w=0,m=0;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter Any Sentence");
    String n=sc.nextLine().trim();
    n=n+" ";
    x1=n;
    int b[]=new int[1000];
    String a[]=new String[1000];
    String c[]=new String[1000];
    int l=x1.length();
    char ch1=x1.charAt(l-1);
    if(n.charAt(l-2)!='.')
    {
        System.out.println("INVALID");
    }
    else
    {
        System.out.println(n);
        n=n.substring(0,n.length()-2)+" ";
    for(int i=0;i<l-1;i++)
    {
        char ch=n.charAt(i);
        if(ch!=' ')
        {
            x=x+ch;
        }
        else
        {
            int q=x.length();
            b[s]=q;
            s++;
```

```
            a[p]=x;
            p++;
            x="";
        }
    }
        for(int i=0;i<s;i++)
    {
        for(int j=0;j<s-i-1;j++)
        {
            if(b[j]>b[j+1])
            {
                int temp=b[j];
                b[j]=b[j+1];
                b[j+1]=temp;

                String temp1=a[j];
                a[j]=a[j+1];
                a[j+1]=temp1;
            }
        }
    }
      for(int i=0;i<s;i++)
       {
        m=b[i];
        int r1=0;
        if(b[i]!=b[i+1])
        {
            System.out.print(a[i]+" ");
        }
        else
        {
             for(int j=i;j<s;j++)
            {
                c[r1]=a[j];
                r1++;
            }
            for(int k=0;k<r1;k++)
            {
                for(int j=0;j<r1-k-1;j++)
                {
                    if(c[j].compareTo(c[j+1])>0)
                    {
                        String temp=c[j];
                        c[j]=c[j+1];
                        c[j+1]=temp;
                    }
                }
```

```
        }
        for(int j=0;j<r1;j++)
        {
            System.out.print(c[j]+" ");
        }
        i=i+r1-1;
 }
}
}
}
}
```

Output:



Conclusion:

Thus by using the above code I have completed the sorting the String Programs.

Ex.No: 6 **PALINDROME**
Date :

Aim :

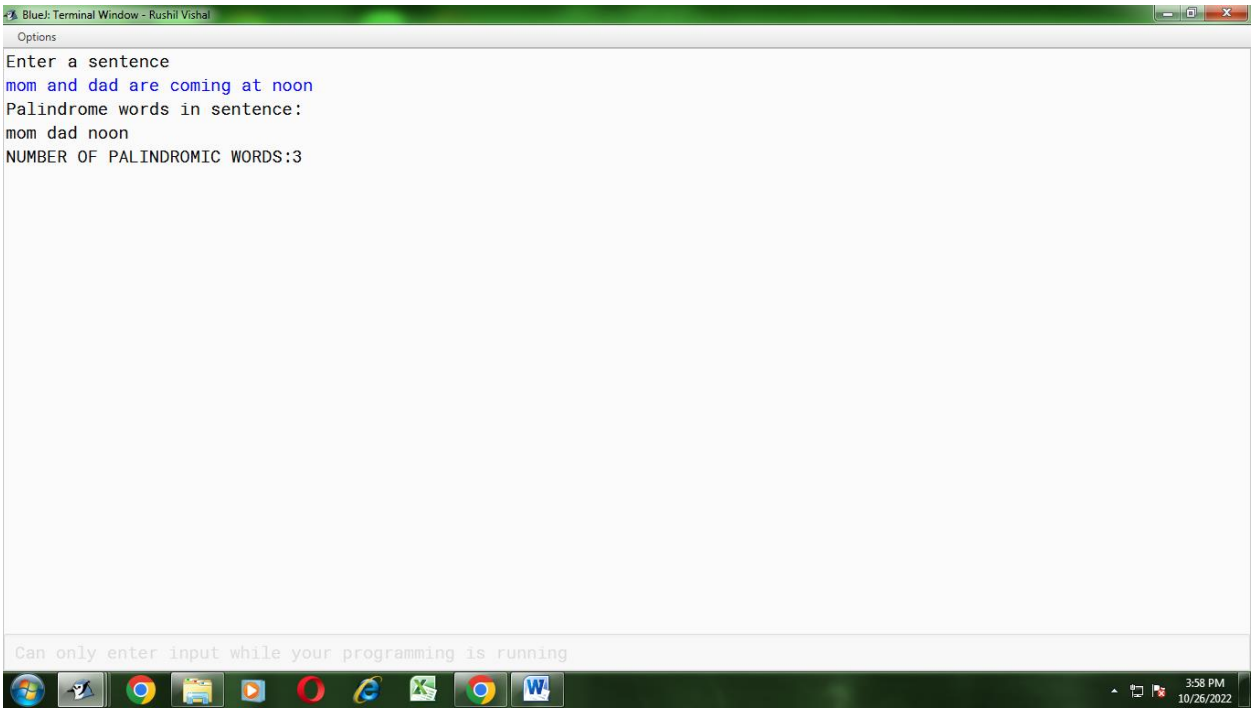
To display the count of palindromic words in the sentence and to display the Palindromic words in the sentence

Source Code:

```
import java.util.Scanner;
public class palindrome
{
    public static void main(String[] args)
    {
        String sen="",wd="",wd1="",palindromeWords="";
        char ch=' ',ch1=' ';
        int i=0,len=0,count=0;
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a sentence");
        sen = sc.nextLine();
        sen=sen+" ";
        sen=sen.toLowerCase();
        len=sen.length();
        for(i=0;i< len;i++)
        {
            ch=sen.charAt(i);
            if(ch==' ')
            {
                if(wd.equals(wd1)==true)
                {
                    palindromeWords+=wd1+" ";
                    count++;
                }
                wd1="";
                wd="";
            }
            else
            {
                wd=wd+ch;
                wd1=ch+wd1;
            }
        }
        if(count>0)
        {
            System.out.println("Palindrome words in sentence:");
```

```
        System.out.println(palindromeWords);

        System.out.println("NUMBER OF PALINDROMIC WORDS:"+count);
    }
    else
    {
        System.out.println("NO PALINDROMIC WORDS FOUND");
    }  }}
```

Output:



Conclusion:

Thus we have displayed the count of palindromic words in the sentence and also the Palindromic words in the sentence

Ex.No: 7 **Composite Magic Number**
Date:

Aim:

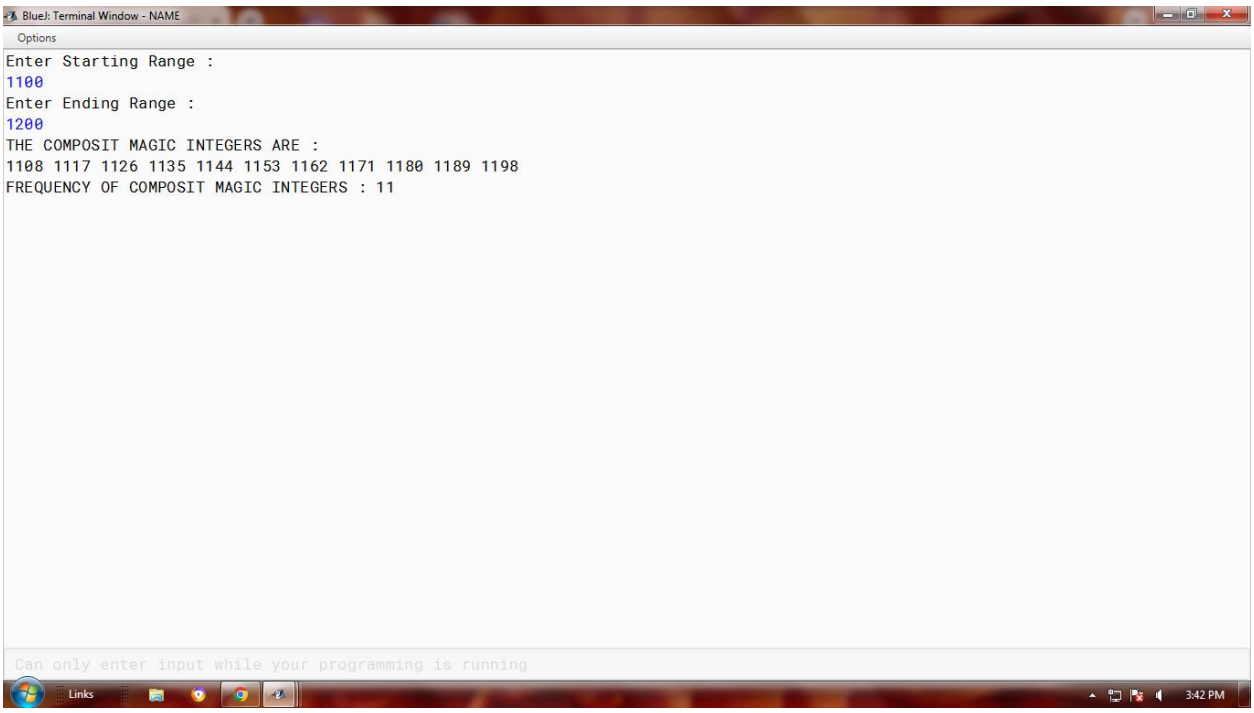
To display the composite magic numbers between the given number limit

Source code:

```
import java.util.*;
public class Composit_Magic
{int i,f=0,r=0;
int isComposit(int a)
   {
      for(i=1;i<=a;i++)
      {
         if(a%i==0)
         f++;
      }
      if(f>2)
      r=1;
      return r;
   }
int isMagic(int b)
   {
int t,r=0,s=0,l;
      t=b;
for(;t!=0;)
      {
         l=t%10;
         s=s+l;
         t=t/10;
         if((s>9)&&(t==0))
         {
            t=s;    s=0;
         }
      }
      if(s==1)
      r=1;
      return r;
   }
   public static void main()
   {
      Scanner in=new Scanner(System.in);
Composit_Magic ob=new Composit_Magic();
int m,n,i,f=0;
System.out.println("Enter Starting Range : ");
```

```
        m=in.nextInt();
System.out.println("Enter Ending Range : ");
        n=in.nextInt();
        if(m<n)
        {
System.out.println("THE COMPOSITE MAGIC INTEGERS ARE : ");
            for(i=m;i<=n;i++)
            {
                if((ob.isComposit(i)==1)&&(ob.isMagic(i)==1))
                {
System.out.print(i+" ");
                    f++;
                }
            }
System.out.println();
System.out.println("FREQUENCY OF COMPOSIT MAGIC INTEGERS : "+f);
        }
        else
System.out.println("INVALID INPUT");
    }}
```

Output:



Conclusion::

Thus we have found all the composite magic numbers between two number limits.

Ex No: 8 **Deleting a Word from a String**
Date :

Aim:

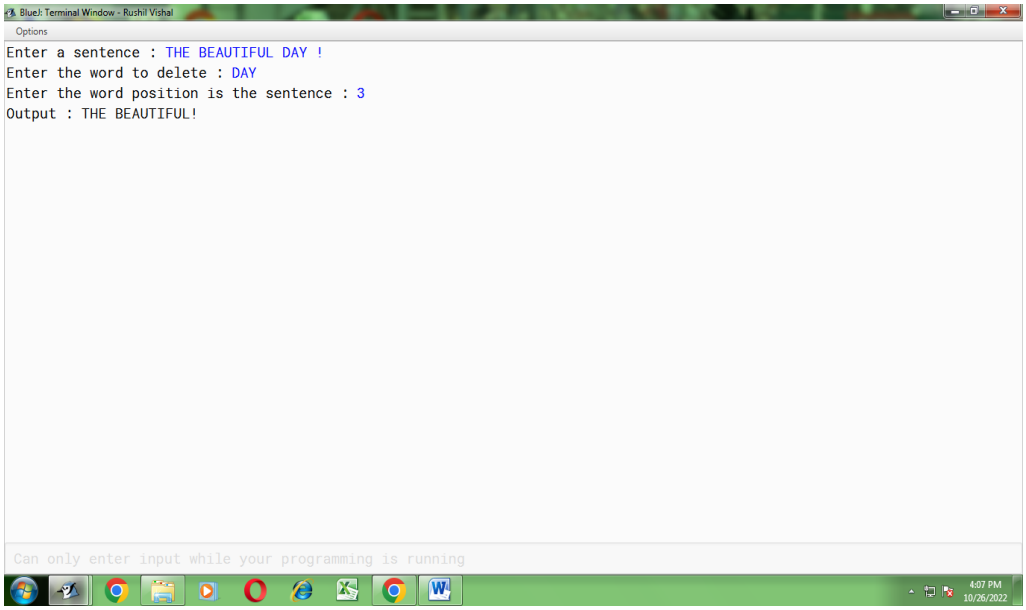
To accept a sentence from the user and delete a word and also reduce the extra blank space from it.

Source Code:

```
import java.util.*;
class RemoveWord
{
    public static void main (String args[])
    {
        Scanner sc = new Scanner(System.in);
       System.out.print("Enter a sentence : ");
        String s = sc.nextLine();
        s = s.toUpperCase();
        int l = s.length();
        char last = s.charAt(l-1);
        if(last != '.' && last != '?' && last != '!')
        {
            System.out.println("Invalid Input. End a sentence with either '.', '?' or '!' only");
        }
        else
        {
            StringTokenizer str = new StringTokenizer(s," .?!");
            int c = str.countTokens();
            String w="",ans = "";
            System.out.print("Enter the word to delete : ");
            String del = sc.next();
            System.out.print("Enter the word position is the sentence : ");
            int x = sc.nextInt();
            if(x<1 || x>c)
            {
                System.out.println("Sorry! The word position entered is out of range");
            }
            else
            {
                for(int i=1; i<=c; i++)
                {
                    w = str.nextToken();
                    if(w.equals(del)==true && i == x)
                        continue;
                    ans = ans + w + " ";
                }
```

```
            System.out.print("Output : "+ans.trim()+last);
        }
    }
} }
```

Output:



Conclusion:

Thus we have deleted a word from a given string and reduced space in a string

Ex.No: 9 **Symmetry Matrix**
Date:

Aim:

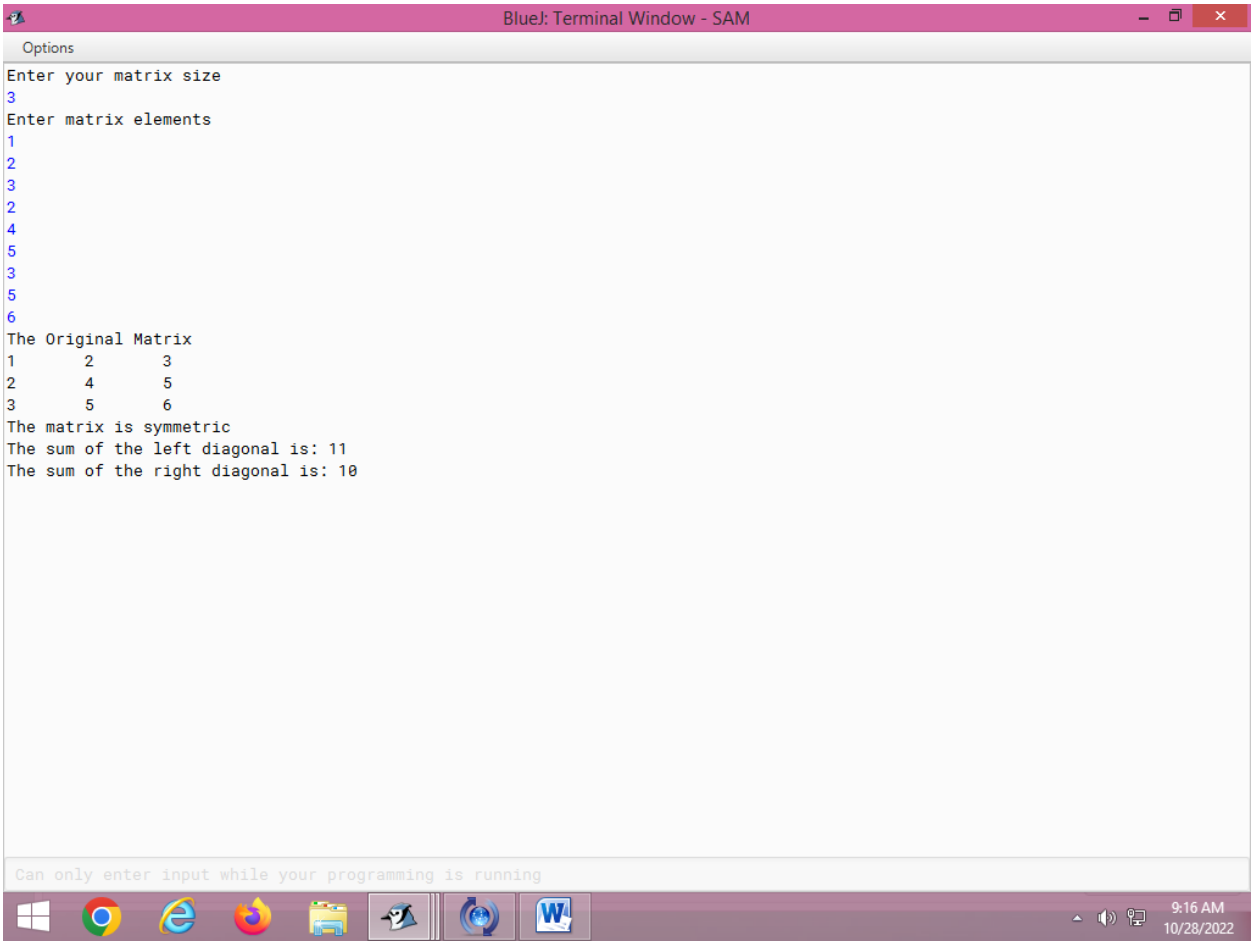
To check whether the matrix is symmetric and to print the sum of both the diagonals.

Source Code:

```
import java.util.*;
public class matrix
{  public static void main (String args[])
   {
      int m; int left=0; int right=0; int k=0;int i; int j;
       Scanner sc = new Scanner(System.in);
       System.out.println("Enter your matrix size");
       m= sc.nextInt();
       if((m>2)&&(m<10))
      {
        int arr[][] = new int[m][m];
        int trans[][] = new int[m][m];
        System.out.println("Enter matrix elements");
       for( i=0; i<m; i++)
       {
          for( j=0; j<m; j++)
          {   arr[i][j]=sc.nextInt();  }
       }
       System.out.println("The Original Matrix");
       for( i=0; i<m; i++)
       {
          for( j=0; j<m; j++)
          {
              System.out.print(arr[i][j] + "\t");
          }
          System.out.println(" ");
       }
       for( i=0; i<m; i++)
       {
          for( j=0; j<m; j++)
         {
          if(arr[i][j]!=arr[j][i])
           {
               k++; }
          }
       }
         if(k>=1)
```

```
        System.out.println("The matrix is not symmetric");
        else
        System.out.println("The matrix is symmetric");
      for(i=0;i<m;i++)
        {
          left=left+arr[i][i];
          right=right+arr[i][m-1-i];
        }
          System.out.println("The sum of the left diagonal is: "+left);
          System.out.println("The sum of the right diagonal is: "+right);
      }
      else
       System.out.println("INVALID MATRIX SIZE");
  }
}
```

OUTPUT:



Conclusion:

Thus we have checked whether the given matrix is symmetric or not and also printed the sum of diagonals of the matrix

Ex.No: 10 **Quiz Program**
Date:

Aim:

To design a program to accept the number of participants and display the scores of each participant by matching their answer with the correct answer.

Source Code:

```
import java.util.*;
public class mcq
{
public static void main(String args[])
{int i,j;
Scanner sc=new Scanner(System.in);
System.out.println("enter the number of participants");
int n= sc.nextInt();
char arr[][]= new char [n][5];
System.out.println("Enter the choices of all participants of 5 subjects");
for(i=0;i<n;i++)
{
for(j=0;j<5;j++)
{arr[i][j]=sc.next().charAt(0);}
System.out.println("Enter the mark of other participant");}
System.out.println("Enter the correct key");
char real[]=new char[5];
for(j=0;j<5;j++)
{real[j]=sc.next().charAt(0);}
int count=0 ,b=1;
for(i=0;i<n;i++)
{
for(j=0;j<5;j++)
{if(arr[i][j]== real[j])
 count=count+1;
}
System.out.println("The marks of participant "+b+" is="+count);
count=0;b=b+1;
}}
}
```

OUTPUT:

```
enter the number of participants
4
Enter the choices of all participants of 5 subjects
d
a
b
c
c
Enter the mark of other participant
a
a
d
b
c
Enter the mark of other participant
b
a
c
d
b
Enter the mark of other participant
d
a
d
c
b
Enter the mark of other participant
Enter the correct key
b
c
d
a
a
The marks of participant 1 is=0
The marks of participant 2 is=1
The marks of participant 3 is=1
The marks of participant 4 is=1
```

Conclusion:

Thus we have designed a program to accept the number of participants and display the scores of each participant by matching their answer with the correct answer.

Ex.No:11 **Octal to Decimal**
Date:

Aim :

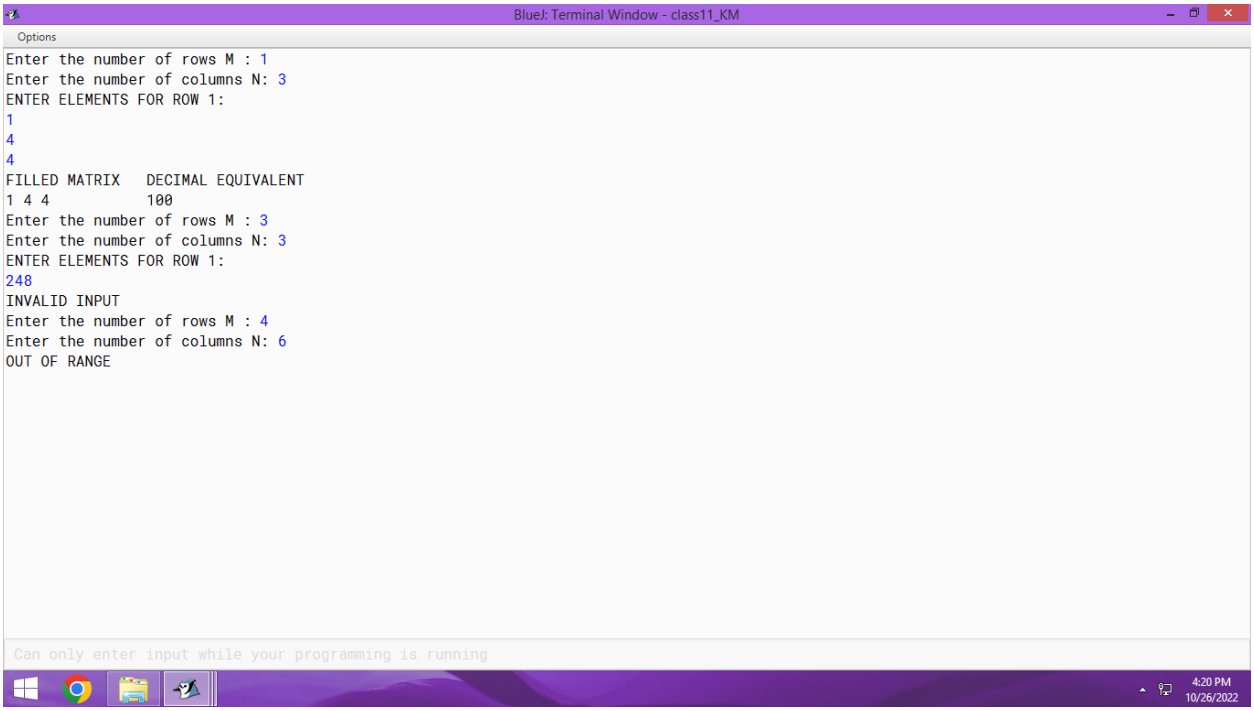
To convert an Octal number to its decimal equivalent.

Source Code :

```
import java.util.*;
public class Oct_to_dec
{
   public static void main(String args[])
   {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter the number of rows M : ");
      int m = sc.nextInt();
      System.out.print("Enter the number of columns N: ");
      int n = sc.nextInt();

      if (m <= 0 || m >= 10 || n <= 2 || n >= 6)
      {
         System.out.println("OUT OF RANGE");
         return;
      }
            int a[][] = new int[m][n];

      for (int i = 0; i < m; i++)
      {
         System.out.println("ENTER ELEMENTS FOR ROW " + (i + 1) + ": ");
         for (int j = 0; j < n; j++)
         {
            a[i][j] = sc.nextInt();
            if (a[i][j] < 0 || a[i][j] > 7) {
               System.out.println("INVALID INPUT");
               return;
            }
         }
      }
            System.out.println("FILLED MATRIX\tDECIMAL EQUIVALENT");
            for (int i = 0; i < m; i++)
      {
         int decNum = 0;
         for (int j = 0; j < n; j++)
         {
            decNum += a[i][j] * Math.pow(8, n - j - 1 );
            System.out.print(a[i][j] + " ");
         }
         System.out.print("\t\t" + decNum);
```

```
            System.out.println();
        }
    }
}
```

Output :



Conclusion:

Thus we haveconverted an octal number to its decimal equivalent.

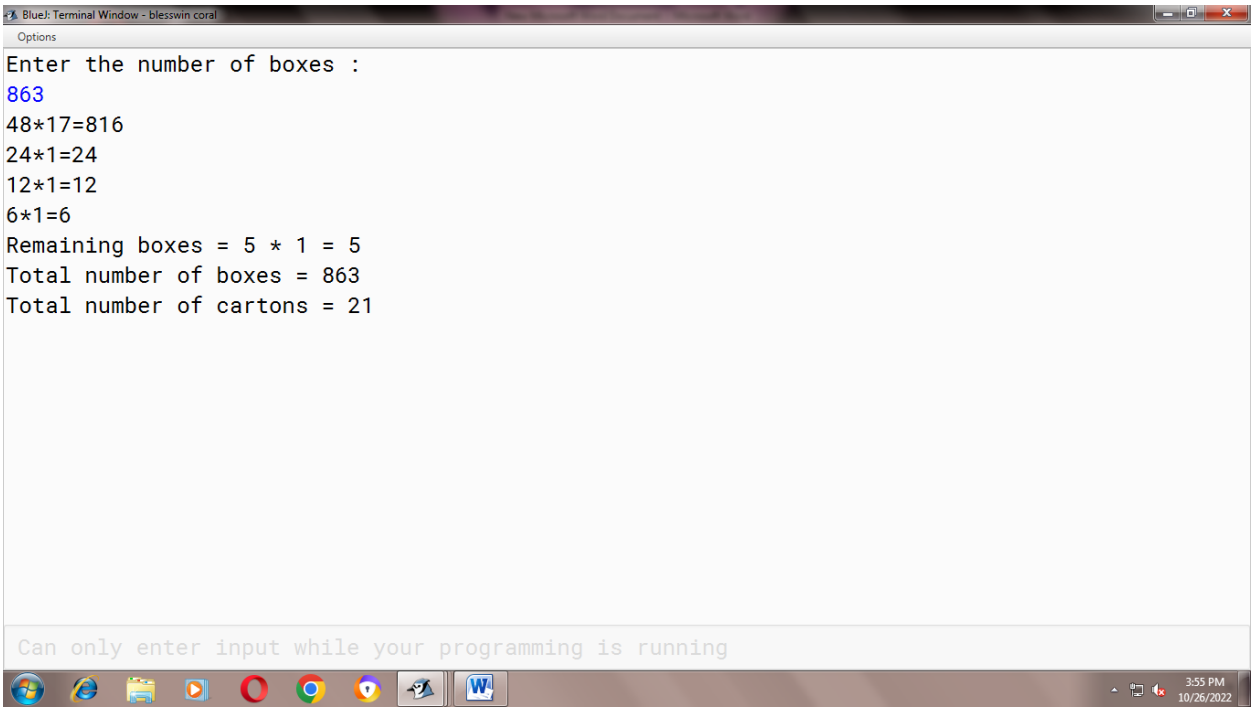
Ex.No:12 **Calculation of Cartons manufactures**
Date:

Aim: To write a program that separates number of boxes into cartons and displays the total number of cartons

Source code:

```
import java.util.Scanner;
class carton{
public static void main(String args[]){
int cartsize[]={48,24,12,6};
System.out.println("Enter the number of boxes : ");
Scanner sc = new Scanner(System.in);
int x= sc.nextInt();
int n=x;
int total=0;
if (x>1000)
System.out.println("Invalid input.");
for (int i=0;i<cartsize.length;i++){
int cartnum=n/cartsize[i];
n=n%cartsize[i];
total +=cartnum;
if (cartnum !=0){
System.out.println(cartsize[i] + "*"+cartnum+"=" +(cartnum*cartsize[i]));
}
}
if (n != 0) {
System.out.println("Remaining boxes = " + n
+ " * 1 = " + n);
total++;
}
else {
System.out.println("Remaining boxes = 0");
}
System.out.println("Total number of boxes = " + x);
System.out.println("Total number of cartons = " + total);
}
}
```

Output:



Conclusion :

Thus by using the above code we have separated number of boxes into cartons and displayed the total number of cartons**.**

Ex.No: 13 **Binary Search**
Date:

Aim:

To seach the admission number of a student using binary search

Source code:

```
import java.util.*;
public class Admission{
   int admnno[];
   public Admission()
   {
      admnno=new int[5];
   }
   public void fillArray()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter admission number in ascending order:");
      for(int i=0;i<admnno.length;i++)
      {
         admnno[i]=sc.nextInt();
      }
   }
   public int binSearch(int l, int u, int v){
      int m=(l+u)/2;
      if(l>u)
         return-1;
      else if(v==admnno[m])
      return 1;
      else
      if(v>admnno[m])
      return binSearch(m+1, u, v);
      else
      return binSearch(l, m-1, v);
   }
   public static void main(String args[])
   {
      Scanner sc = new Scanner(System.in);
      Admission obj = new Admission();
      obj.fillArray();
      System.out.print("Enter the key to search:");
      int key = sc.nextInt();
      int result = obj .binSearch(0, 4, key);
      if(result == 1)
         System.out.println(key+" is available.");
```

```
        else
            System.out.println(key +" is unavailable.");
    }
}
```

Output:

Conclusion:

Thus we have successfully searched the admission number of the student using binary search.

Ex.No:14 **Reversing a String**
Date:

Aim : To reverse a given sentence.

Source Code :

```
import java.util.*;

public class string{
   int len;

   String sent,new_sent,rev;
   string(){
      sent="";
      new_sent="";
      rev="";
   }

   public void input(){
      Scanner sc = new Scanner(System.in);
      System.out.println("ENTER THE SENTENCE:");
      sent = sc.nextLine();
   }

   public void reverse(){
      len=sent.length();
      for (int i=len-1;i>=0;--i){

         rev=rev+(sent.charAt(i));

   }
}

   public void display(){
      System.out.println("Original string ");
      System.out.println(sent);
      System.out.println("reversed  string ");

      System.out.println(rev);
   }

   public static void main(String args[]){
      string ob=new string();
      ob.input();
      ob.reverse();
      ob.display();       }     }
```

Output :

```
BlueJ: Terminal Window - Kevin
Options
ENTER THE SENTENCE:
THE BIRD IS FLYING
Original string
THE BIRD IS FLYING
reversed  string
GNIYLF SI DRIB EHT
```

Conclusion : Thus by using the above source code the given string has been reversed

Ex.No : 15 **Frequency of Capital Letter**
Date :

Aim :

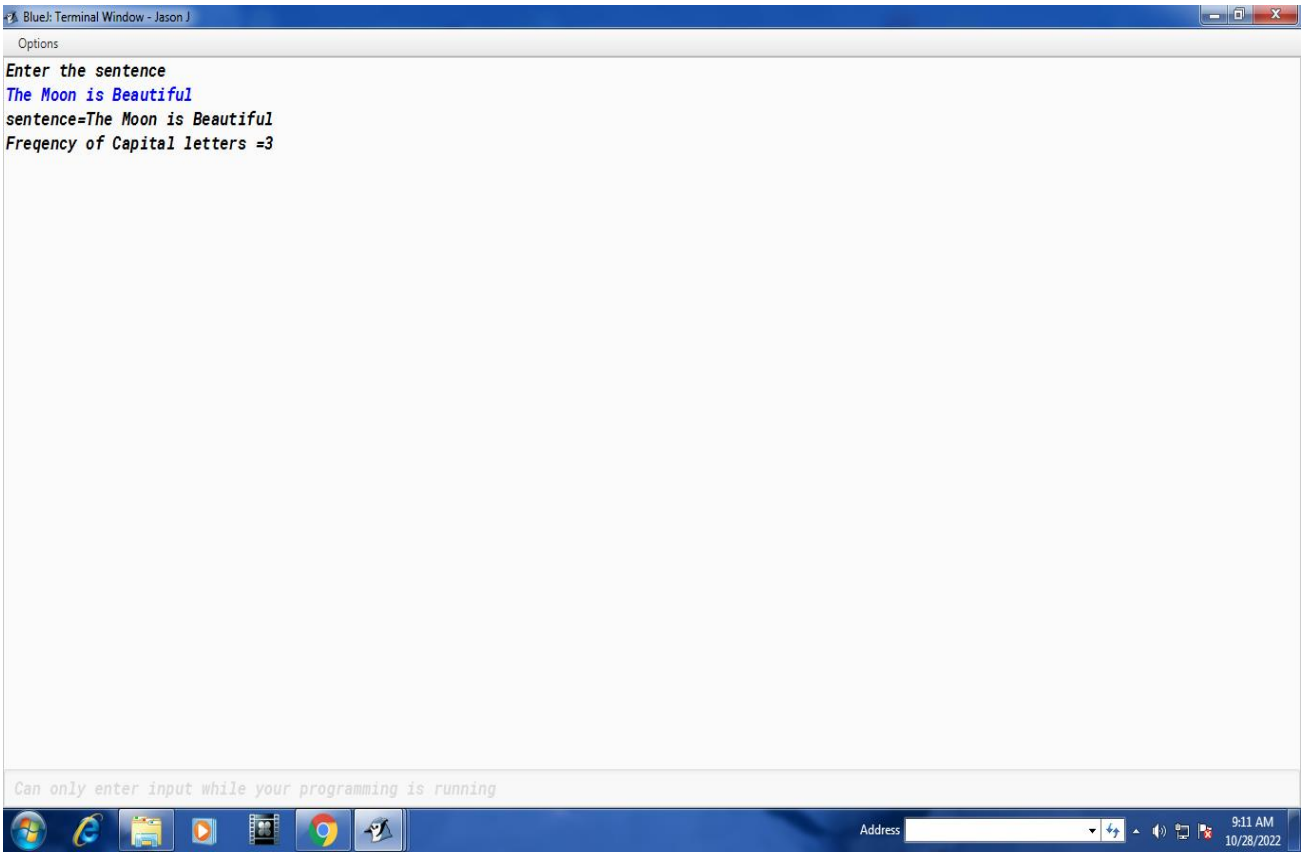
To check whether a sentence has words beginning with a capital letter or not, and count the frequency of word beginning capital with capital letters.

Source Code :

```
import java.util.*;
class Capital
{
   String sent;
   int freq;
   Capital()
   {
     sent="";
     freq=0;
   }
   void input()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the sentence");
      sent=sc.nextLine();
   }
   boolean isCap(String w)
   {
       char c=w.charAt(0);
       if (Character.isUpperCase(c) )
         return true;
       else
         return false;
   }
   void display()
   {
      System.out.println("sentence="+sent);
      String b=" "+sent;
      char c;
      for(int i=0;i<b.length();i++)
      {
         c=b.charAt(i);
         if (c==' ')
         {
            if(Character.isUpperCase(b.charAt(i+1)) )
               freq++;
         }
```

```
        }
        System.out.println("Freqency of Capital letters ="+freq);
    }
    public static void main()
    {
        Capital ob=new Capital();
        ob.input();
        ob.display();
    }
}
```

Output :



Conclusion:

Thus we have found the frequency for the given sentence.

Ex.No: 16
Date: **Date Calculation**

Aim:
To calculate the date after N days.

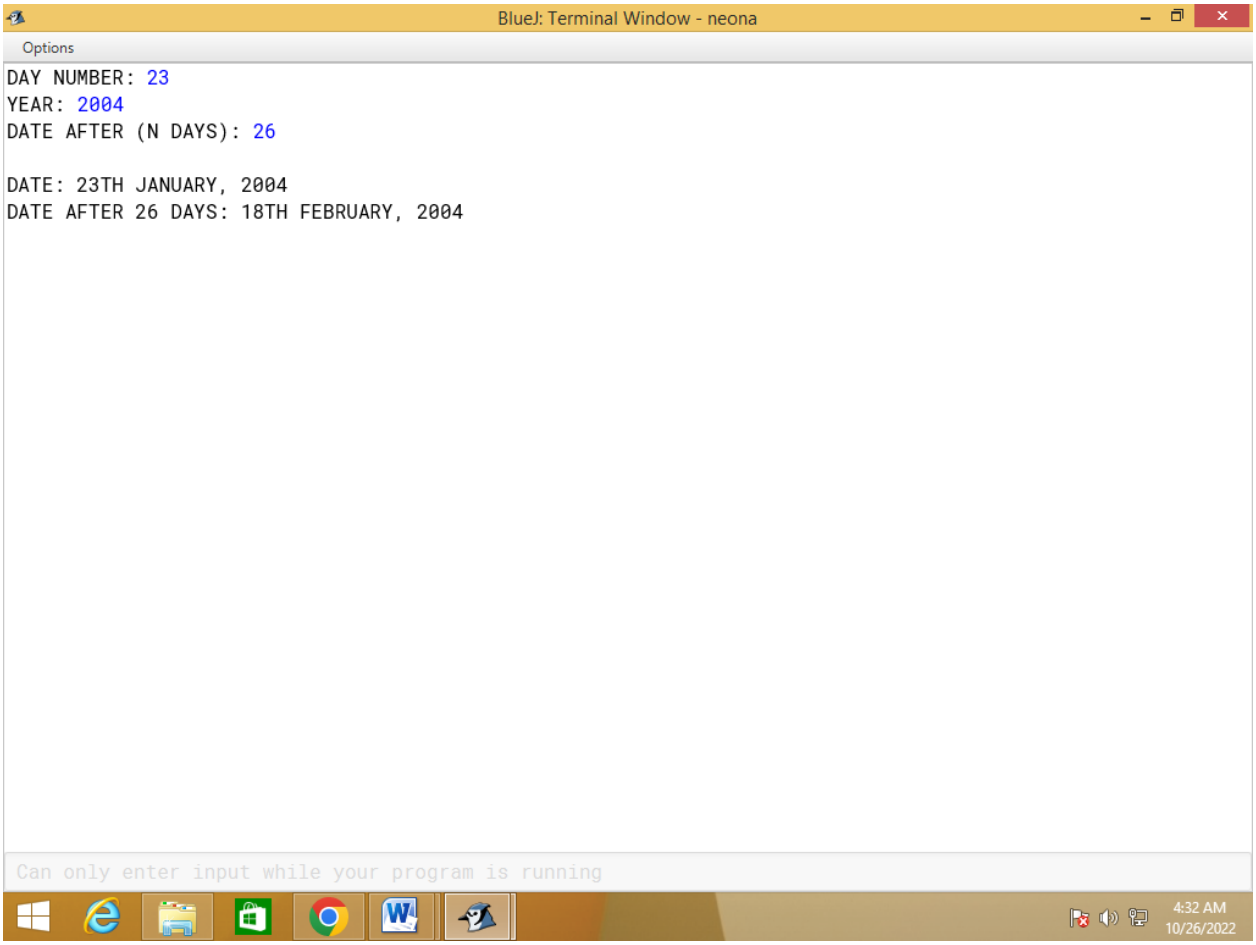
Source Code:

```
import java.util.Scanner;

public class DateCalculator
{
    public static boolean isLeapYear(int y) {
        boolean ret = false;

        if (y % 400 == 0) {
            ret = true;
        }
        else if (y % 100 == 0) {
            ret = false;
        }
        else if (y % 4 == 0) {
            ret = true;
        }
        else {
            ret = false;
        }

        return ret;
    }

    public static String computeDate(int day, int year) {
        int monthDays[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
        String monthNames[] = {"JANUARY", "FEBRUARY", "MARCH", "APRIL", "MAY",
"JUNE",  "JULY", "AUGUST", "SEPTEMBER",
                                "OCTOBER", "NOVEMBER", "DECEMBER"};

        boolean leap = isLeapYear(year);

        if (leap) {
            monthDays[1] = 29;
        }

        int i = 0;
        int daySum = 0;
```

```
        for (i = 0; i < monthDays.length; i++) {
            daySum += monthDays[i];
            if (daySum >= day) {
                break;
            }
        }
        int date = day + monthDays[i] - daySum;

        StringBuffer sb = new StringBuffer();
        sb.append(date);
        sb.append("TH ");
        sb.append(monthNames[i]);
        sb.append(", ");
        sb.append(year);

        return sb.toString();
    }

    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        System.out.print("DAY NUMBER: ");
        int dayNum = in.nextInt();
        System.out.print("YEAR: ");
        int year = in.nextInt();
        System.out.print("DATE AFTER (N DAYS): ");
        int n = in.nextInt();

        if (dayNum < 1 || dayNum > 366) {
            System.out.println("DAY NUMBER OUT OF RANGE");
            return;
        }

        if (n < 1 || n > 100) {
            System.out.println("DATE AFTER (N DAYS) OUT OF RANGE");
            return;
        }

        String dateStr = computeDate(dayNum, year);

        int nDays = dayNum + n;
        int nYear = year;
        boolean leap = isLeapYear(year);

        if (leap && nDays > 366) {
            nYear = nYear + 1;
            nDays = nDays - 366;
        }
```

```
        else if (nDays > 365) {
            nYear = nYear + 1;
            nDays = nDays - 365;
        }

        String nDateStr = computeDate(nDays, nYear);

        System.out.println();
        System.out.println("DATE: " + dateStr);
        System.out.println("DATE AFTER " + n
                    + " DAYS: " + nDateStr);
    }

}
```

OUTPUT:



CONCLUSION:

Thus we have displayed the date, date after N days .

Ex.No: 17 **PrimeAdam**
Date:

Aim:

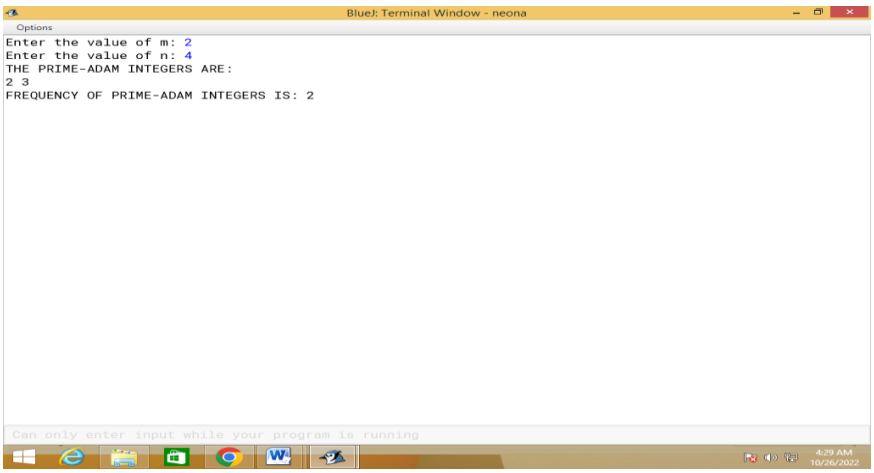
To check whether the given number is Prime and Adam,

Source Code:

```
import java.util.Scanner;

public class PrimeAdam
{
    public static int reverse(int num) {
        int rev = 0;

        while (num != 0) {
            int d = num % 10;
            rev = rev * 10 + d;
            num /= 10;
        }

        return rev;
    }

    public static boolean isAdam(int num) {
        int sqNum = num * num;
        int revNum = reverse(num);
        int sqRevNum = revNum * revNum;
        int rev = reverse(sqNum);

        return rev == sqRevNum;
    }

    public static boolean isPrime(int num) {
        int c = 0;

        for (int i = 1; i <= num; i++) {
            if (num % i == 0) {
                c++;
            }
        }

        return c == 2;
    }

    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
```

```
        System.out.print("Enter the value of m: ");
        int m = in.nextInt();
        System.out.print("Enter the value of n: ");
        int n = in.nextInt();
        int count = 0;

        if (m >= n) {
            System.out.println("INVALID INPUT");
            return;
        }

        System.out.println("THE PRIME-ADAM INTEGERS ARE:");
        for (int i = m; i <= n; i++) {
            boolean adam = isAdam(i);
            if (adam) {
                boolean prime = isPrime(i);
                if (prime) {
                    System.out.print(i + " ");
                    count++;
                }    }   }

                if (count == 0) {
            System.out.print("NIL");
        }

        System.out.println();
        System.out.println("FREQUENCY OF PRIME-ADAM INTEGERS IS: " + count);
    }       }
```

OUTPUT:



CONCLUSION

Thus we have displayed the frequency of Prime-Adam integers.

Ex.No: 18

Date: **Matrix Sorting**

Aim:
To find the diagonal of a matrix using sorting.

Source Code :

```
import java.util.Scanner;
public class Gerwin
{
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter matrix size");
        int m = in.nextInt();
        if (m <= 3 || m >= 10) {
            System.out.println("size is invalid.");
            return;
        }
        int a[][] = new int[m][m];
        System.out.println("Enter elements");
        for (int i = 0; i < m; i++) {
            System.out.println("Enter rows " + (i+1) + ":");
            for (int j = 0; j < m; j++) {
                a[i][j] = in.nextInt();
                if (a[i][j] < 0) {
                    System.out.println("Invalid size");
                    return;
                }
            }
        }
        System.out.println("Original matrix");
        printMatrix(a, m);
        sortNonBoundaryMatrix(a, m);
        System.out.println("Rearranged Matrix");
        printMatrix(a, m);
        computePrintDiagonalSum(a, m);
    }

    public static void sortNonBoundaryMatrix(int a[][], int m) {
        int b[] = new int[(m - 2) * (m - 2)];
        int k = 0;
        for (int i = 1; i < m - 1; i++) {
            for (int j = 1; j < m - 1; j++) {
                b[k++] = a[i][j];
            }   }
                for (int i = 0; i < k - 1; i++) {
            for (int j = 0; j < k - i - 1; j++) {
```

```
                if (b[j] > b[j + 1]) {
                    int t = b[j];
                    b[j] = b[j+1];
                    b[j+1] = t;
                }
            }
        }
        k = 0;
        for (int i = 1; i < m - 1; i++) {
            for (int j = 1; j < m - 1; j++) {
                a[i][j] = b[k++];
            }
        }
    }
    public static void computePrintDiagonalSum(int a[][], int m) {
        int sum = 0;
        System.out.println("Diagonal elements");
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < m; j++) {
                if (i == j || i + j == m - 1) {
                    sum += a[i][j];
                    System.out.print(a[i][j] + "\t");
                }
                else {
                    System.out.print("\t");
                }
            }
            System.out.println();
        }
        System.out.println("Suma of diagonls = " + sum);
    }

    public static void printMatrix(int a[][], int m) {
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < m; j++) {
                System.out.print(a[i][j] + "\t");
            }
            System.out.println();
        }
    }
}
```

Output :

```
Enter matrix size4
Enter elements
Enter rows 1:
12
1
2
4
Enter rows 2:
5
4
8
9
Enter rows 3:
8
7
3
5
Enter rows 4:
8
78
54
25
```

```
Original matrix
12      1       2       4
5       4       8       9
8       7       3       5
8       78      54      25
Rearranged Matrix
12      1       2       4
5       3       4       9
8       7       8       5
8       78      54      25
Diagonal emlements
12                      4
        3       4
        7       8
8                       25
Suma of diagonls = 71
```

Conclusion:

Thus by using the above code we have found the diagonal of a matrix using sorting.

Ex.No: 19
Date: **Mirror Matrix**

Aim:

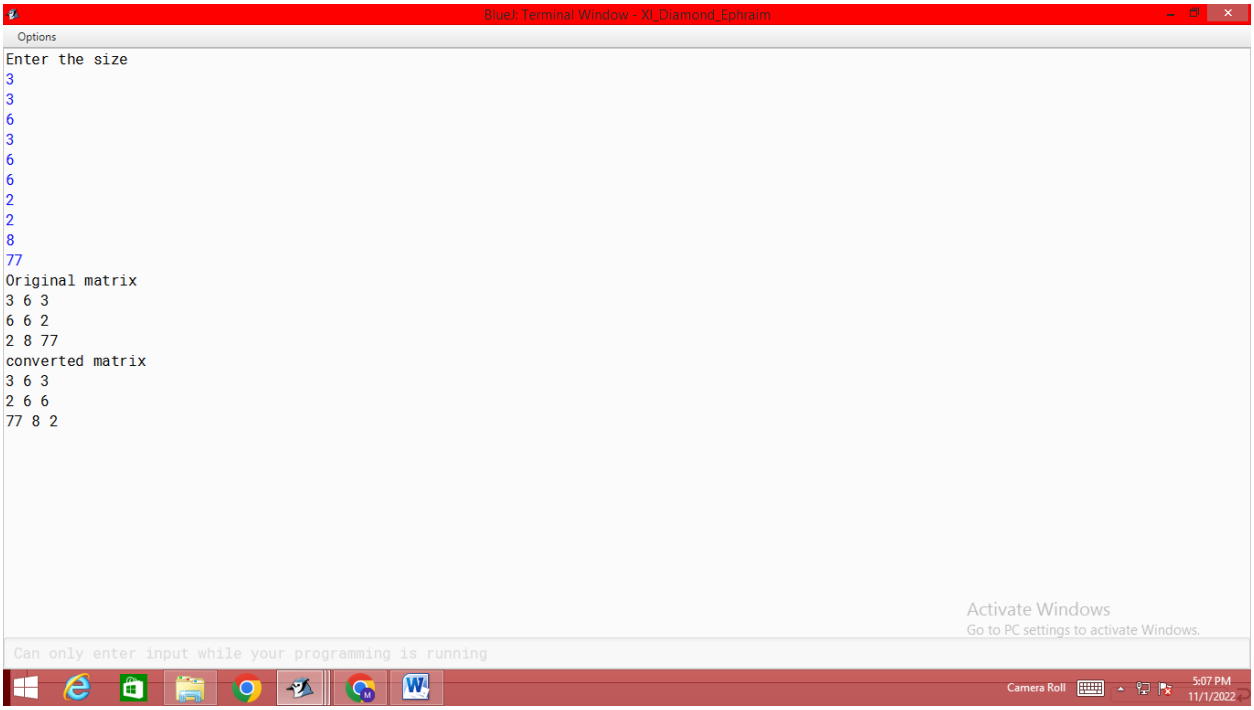
To display the mirror image of the given matrix,

Source code:

```
import java.util.*;
class mirror
{
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter the size");
int m=sc.nextInt();
if(m<2||m>20)
{
System.out.println("invalid size");
}
else
{
int b[][]=new int[m][m];
int a[][]=new int[m][m];
for(int i=0;i<m;i++)
{
for(int j=0;j<m;j++)
{
a[i][j]=sc.nextInt();
}
}
for(int i=0;i<m;i++)
{ int k=2;
for(int j=0;j<m;j++)
{
b[i][j]=a[i][k];
k--;
}
}
System.out.println("Original matrix");
for(int i=0;i<m;i++)
{
for(int j=0;j<m;j++)
{
System.out.print(a[i][j]+" ");
```

```
}
System.out.println();
}
System.out.println("converted matrix");
for(int i=0;i<m;i++)
{
for(int j=0;j<m;j++)
{
System.out.print(b[i][j]+" ");
}
System.out.println();
}
}
}
}
```

Output:



Conclusion:

Thus we have displayed the mirror image of the given matrix,

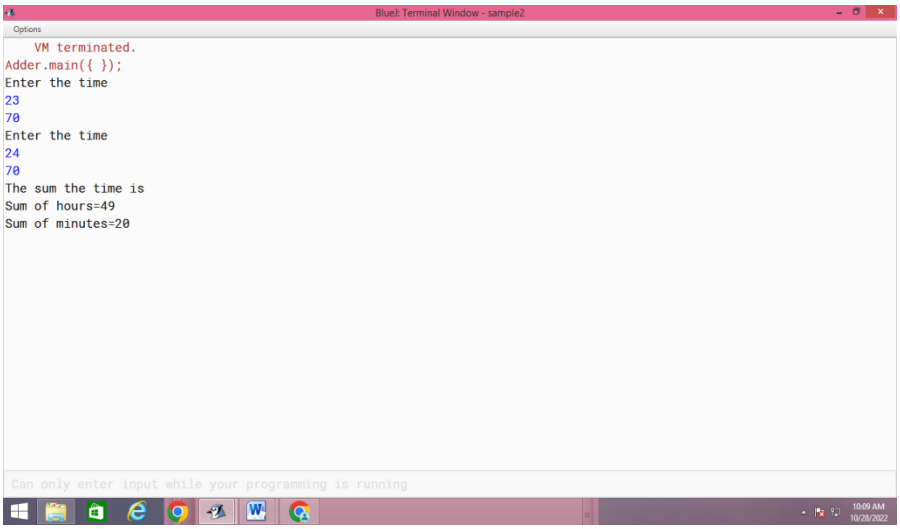
Ex.No: 20 **Matrix--Adder**
Date:

Aim :To add two accepted time and display them in separate as hours and minutes

Source code:

```
import java.util.*;
public class Adder{
int a[];
Adder() {
a = new int[2];
}
void readtime() {
Scanner sc=new Scanner(System.in);
System.out.println("Enter the time ");
for (int i=0;i<2;i++){
a[i]=sc.nextInt();
}}
void addtime(Adder X, Adder Y) {
int hour1 = X.a[0];
int min1 = X.a[1];
int hour2 = Y.a[0];
int min2 = Y.a[1];
int hourSum = hour1 + hour2;
int minSum = min1 + min2;
int cm=(minSum/60);
a[0] = hourSum + cm;
a[1] = minSum%60;
}
    void disptime(){
   System.out.println("The sum the time is ");
    System.out.println("Sum of hours="+a[0]);
    System.out.println("Sum of minutes="+a[1]);
              }
public static void main(String[]args){
  Adder obj1=new Adder();
  Adder obj2=new Adder();
   Adder sumobj=new Adder();
  obj1.readtime();
  obj2.readtime();
sumobj.addtime(obj1,obj2);
sumobj.disptime();
              }}
```

Output:



Conclusion:

To add two accepted time and display them in separate as hours and minutes