**This material contains some of the most important Programs for ICSE. The programs have been solved & cross checked.**

**Question:**

Write a Program in Java to input a number and check whether it is a **Disarium Number** or not.

**Note:** A number will be called DISARIUM if sum of its digits powered with their respective position is equal to the original number.

**For example** 135 is a DISARIUM  
(Workings 11+32+53 = 135, some other **DISARIUM** are 89, 175, 518 etc)

|  |
| --- |
| import java.io.\*;  class Disarium      {      public static void main(String[] args)throws IOException          {              BufferedReader br=new BufferedReader (new InputStreamReader(System.in));              System.out.print("Enter a number : ");              int n = Integer.parseInt(br.readLine());              int copy = n, d = 0, sum = 0;              String s = Integer.toString(n); //converting the number into a String              int len = s.length(); //finding the length of the number i.e. no.of digits                while(copy>0)              {                  d = copy % 10; //extracting the last digit                  sum = sum + (int)Math.pow(d,len);                  len--;                  copy = copy / 10;              }                if(sum == n)                  System.out.println(n+" is a Disarium Number.");              else                  System.out.println(n+" is not a Disarium Number.");          }      } |

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Question:**

Define a class named **movieMagic** with the following description:

**Instance variables/data members:**

int year            -           to store the year of release of a movie

String title       -           to store the title of the movie.

float rating      -           to store the popularity rating of the movie. (minimum rating = 0.0 and maximum rating = 5.0)

**Member Methods:**

(i)         movieMagic()              Default constructor to initialize numeric data members to 0 and String data member to “”.

(ii)        void accept()               To input and store year, title and rating.

(iii)       void display()              To display the title of a movie and a message based on the rating as per the table below.

|  |  |
| --- | --- |
| **Rating** | **Message to be displayed** |
| 0.0 to 2.0 | Flop |
| 2.1 to 3.4 | Semi-hit |
| 3.5 to 4.5 | Hit |
| 4.6 to 5.0 | Super Hit |

Write a main method to create an object of the class and call the above member methods.

import java.io.\*;

class movieMagic

    {

     int year;

     String title;

     float rating;

movieMagic() // default constructor

    {

     year = 0;

     rating = 0.0f; // notice the 'f'

     title = "";

    }

void accept() throws IOException

    {

     BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

     System.out.print("Enter the title of the movie : ");

     title = br.readLine();

     System.out.print("Enter the year of its release : ");

     year = Integer.parseInt(br.readLine());

     System.out.print("Enter the movie rating : ");

     rating = Float.parseFloat(br.readLine());

    }

void display()

    {

     System.out.println("The title of the movie is : "+title);

     if( rating >= 0.0 && rating <= 2.0 ) {

      System.out.println("The movie was a Flop");

     }  else if( rating >= 2.1 && rating <= 3.4 ) {

      System.out.println("The movie was a Semi-hit");

     }  else if( rating >= 3.5 && rating <= 4.5 ) {

      System.out.println("The movie was a Hit");

     }  else   {

      System.out.println("The movie was a Super Hit");

     }

    }

public static void main(String args[]) throws IOException

    {

     movieMagic ob = new movieMagic(); // creating object of the class movieMagic

     ob.accept();

     ob.display();

    }

    }

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Question:**

Write a program in Java to accept a number and check whether it belongs to the Fibonacci Series (sequence) or not.

**Fibonacci Series:**  
The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …

The first two numbers in the series is ’0′ and ’1′ and every next number is found by adding up the two numbers before it.

The 2 is found by adding the two numbers before it (1+1)  
Similarly, the 3 is found by adding the two numbers before it (1+2),  
And the 5 is (2+3),  
and so on!

**Example:** the next number in the sequence above would be 21+34 = 55  
It is that simple!

Here is a longer list:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, …

import java.io.\*;

class IsFibonacci

{

public static void main(String args[])throws IOException

   {

     BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

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

     int n = Integer.parseInt(br.readLine());

     if(n<0)

        System.out.println("Kindly enter a positive number.");

     else

     {

         int a=0, b=1 ,c=0;

         /\* 'a' is the 1st term, 'b' is the 2nd term and 'c' is the 3rd term

          \* 'c' stores the last generated term of the Fibonacci series \*/

          while(c<n) // Loop goes on till the 3rd term is less than the given number

          {

              c = a + b; // Generating the terms of Fibonacci Series

              a = b;

              b = c;

          }

          /\* When the control comes out of the while loop, either the

           \* 3rd term is equal to the number or greater than it \*/

           if(c==n) // If the last term = number, then it belongs to Fibonacci Series

              System.out.println("Output : The number belongs to Fibonacci Series.");

           else

              System.out.println("Output : The number does not belong to Fibonacci Series.");

     }

   }

}

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### Question:

A special two-digit number is such that when the sum of the digits is added to the product of its digits, the result is equal to the original two-digit number.

**Example:**  
Consider the number 59.Sum of digits = 5+9=14  
Product of its digits = 5 x 9 = 45  
Sum of the digits and product of digits = 14 + 45 = 59

Write a program to accept a two-digit number. Add the sum of its digits to the product of its digits. If the value is equal to the number input, output the message “special-two digit number” otherwise, output the message “Not a special two-digit number”.

|  |
| --- |
| import java.io.\*;  class Special\_Q5\_ICSE2014      {      public static void main(String args[])throws IOException          {              BufferedReader br=new BufferedReader (new InputStreamReader(System.in));              System.out.print("Enter a 2 digit number : ");              int n = Integer.parseInt(br.readLine());                int first, last, sum, pro;              if(n<10 || n>99) //Checking whether entered number is 2 digit or not                  System.out.println("Invalid Input! Number should have 2 digits only.");              else                  {                      first = n/10; //Finding the first digit                      last = n%10; //Finding the last digit                      sum = first + last; //Finding the sum of the digits                      pro = first \* last; //Finding the product of the digits                        if((sum + pro) == n)                      {                        System.out.println("Output : The number "+n+" is a Special Two-Digit Number.");                      }                      else                      {                        System.out.println("Output : The number is Not a Special Two-Digit Number.");                      }                  }          }      } |

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Question:**

Write a program to assign a full path and file name as given below. Using library functions, extract and output the file path, file name and file extension separately as shown.

**Input:** C:\Users\admin\Pictures\flowers.jpg  
**Output:**  
Path: C:\Users\admin\Pictures\  
File name: flower  
Extension: jpg

import java.io.\*;

class File\_Q6\_ICSE2014

    {

    public static void main(String args[])throws IOException

        {

            BufferedReader br=new BufferedReader (new InputStreamReader(System.in));

            System.out.print("Enter the full path of the file : ");

            String s = br.readLine();

            int x = s.lastIndexOf('\\'); // Finding position of last backward slash

            int y = s.lastIndexOf('.'); // Finding position of last '.'

            String path = s.substring(0,(x+1));

            String file = s.substring((x+1),y);

            String extn = s.substring((y+1));

            System.out.println("Output :");

            System.out.println("Path : "+path);

            System.out.println("File Name : "+file);

            System.out.println("Extension : "+extn);

        }

    }

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### ****Question:****

Write a Program in Java to input a number and check whether it is an **Automorphic Number** or not.

**Note:** An automorphic number is a number which is present in the last digit(s) of its square.  
**Example:** 25 is an automorphic number as its square is 625 and 25 is present as the last digits

import java.io.\*;

class Automorphic

{

public static void main(String args[]) throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter a Number : "); // Inputting the number

int n = Integer.parseInt(br.readLine());

int sq = n\*n; // Finding the square

int c = 0, copy = n;

// While loop for counting the number of digits in the number

while(copy > 0)

{

c++;

copy = copy/10;

}

/\* Finding the end digits of the square.

\* If the number has 2 digits, then we need to find last 2 digits of square

\* i.e. do a 'sq % 100' operation

\*/

int end = sq % (int)Math.pow(10,c);

if(n == end) // If the square ends with the number then it is Automorphic

System.out.print(n+" is an Automorphic Number.");

else

System.out.print(n+" is not an Automorphic Number.");

}

}

#### Question:

Write a program to input a word from the user and remove the duplicate characters by replacing the sequence of duplicate characters by its single occurrence.

**Example:**

INPUT – Jaaavvvvvvvvaaaaaaaaaaa  
OUTPUT – Java

INPUT – Heeeiiiissggoiinggg  
OUTPUT – Heisgoing

import java.io.\*;

class RemoveDupChar

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter any word: "); // Inputting the word

String s = br.readLine();

s = s + " "; // Adding a space at the end of the word

int l=s.length(); // Finding the length of the word

String ans=""; // Variable to store the final result

char ch1,ch2;

for(int i=0; i<l-1; i++)

{

ch1=s.charAt(i); // Extracting the first character

ch2=s.charAt(i+1); // Extracting the next character

// Adding the first extracted character to the result if the current and the next characters are different

if(ch1!=ch2)

{

ans = ans + ch1;

}

}

System.out.println("Word after removing repeated characters = "+ans); // Printing the result

}

}

#### Question:

Write a program to input a number. Count and print the frequency of each digit present in that number. The output should be given as:  
**Sample Input:** 44514621  
**Sample Output:**=====================  
Digit             Frequency  
=====================  
1                         2  
2                         1  
4                         3  
5                         1  
6                         1

import java.io.\*;

class Digit\_Freq

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter any number : ");

int n = Integer.parseInt(br.readLine());

int freq[]=new int[10]; //array for storing frequency of all digits

for(int i=0; i<10; i++)

{

freq[i]=0; //intializing the count of every digit with '0'

}

/\*Note: Frequency of digit '0' is stored in freq[0], frequency of '1' in freq[1] and so on\*/

System.out.println("Output:");

System.out.println("===================="); //this is just for styling the look of the output

System.out.println("Digit\tFrequency");

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

int d;

while(n>0)

{

d=n%10; //extracting digit from the end

freq[d]++; //increasing the frequency of that digit.

n=n/10;

}

for(int i=0; i<10; i++)

{

if(freq[i]!=0) //printing only those digits whose count is not '0'

System.out.println(" "+i+"\t "+freq[i]);

}

}

}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### Question:

Write a program to input a string (word). Convert it into lowercase letters. Count and print the frequency of each alphabet present in the string. The output should be given as:  
**Sample Input:** Alphabets  
**Sample Output:**==========================  
Alphabet             Frequency  
==========================  
a                              2  
b                              1  
e                              1  
h                              1  
l                               1  
p                              1  
s                              1  
t                               1

import java.io.\*;

class Alphabet\_Freq

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter any string: ");

String s = br.readLine();

s=s.toLowerCase(); //converting the string into lowercase

int l=s.length(); //finding the length of the string

char alph[]=new char[26]; //array for storing alphabets from 'a' to 'z'

int freq[]=new int[26]; //array for storing frequency of all alphabets

char c='a';

for(int i=0; i<26; i++)

{

alph[i]=c; //storing all alphabets from 'a' till 'z' in alph[] array

freq[i]=0; //intializing the count of every alphabet with '0'

c++;

}

char ch;

System.out.println("Output:");

System.out.println("=========================="); //this is just for styling the look of the output

System.out.println("Alphabet\tFrequency");

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

/\* Counting frequency of alphabets begins below \*/

for(int i=0; i<26; i++)

{

for(int j=0; j<l; j++)

{

ch=s.charAt(j); //extracting characters of the string one by one

if(ch==alph[i]) //first checking the whole string for 'a', then 'b' and so on

freq[i]++; //increasing count of those aplhabets which are present in the string

}

}

for(int i=0; i<26; i++)

{

if(freq[i]!=0) //printing only those alphabets whose count is not '0'

System.out.println(" "+alph[i]+"\t\t "+freq[i]);

}

}

}

Question:

A Smith number is a composite number, the sum of whose digits is the sum of the digits of its prime factors obtained as a result of prime factorization (excluding 1). The first few such numbers are 4, 22, 27, 58, 85, 94, 121 ………………..

Examples:

1. 666

Prime factors are 2, 3, 3, and 37

Sum of the digits are (6+6+6) = 18

Sum of the digits of the factors (2+3+3+(3+7)) = 18

2. 4937775

Prime factors are 3, 5, 5, 65837

Sum of the digits are (4+9+3+7+7+7+5) = 42

Sum of the digits of the factors (3+5+5+(6+5+8+3+7)) = 42

Write a program to input a number and display whether the number is a Smith number or not.

Sample data:

Input 94 Output SMITH Number

Input 102 Output NOT SMITH Number

Input 666 Output SMITH Number

Input 999 Output NOT SMITH Number

import java.io.\*;

class Smith

{

static BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

//function for finding sum of digits

int sumDig(int n)

{

int s=0;

while(n>0)

{

s=s+n%10;

n=n/10;

}

return s;

}

//function for generating prime factors and finding their sum

int sumPrimeFact(int n)

{

int i=2, sum=0;

while(n>1)

{

if(n%i==0)

{

sum=sum+sumDig(i); //Here 'i' is the prime factor of 'n' and we are finding its sum

n=n/i;

}

else

i++;

}

return sum;

}

public static void main(String args[]) throws IOException

{

Smith ob=new Smith();

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

int n=Integer.parseInt(br.readLine());

int a=ob.sumDig(n);// finding sum of digit

int b=ob.sumPrimeFact(n); //finding sum of prime factors

System.out.println("Sum of Digit = "+a);

System.out.println("Sum of Prime Factor = "+b);

if(a==b)

System.out.print("It is a Smith Number");

else

System.out.print("It is Not a Smith Number");

}

}

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### ****Question:****

Write a Program in Java to input a number in Decimal number system and convert it into its equivalent number in the Hexadecimal number system.

**Note:** Hexadecimal Number system is a number system which can represent a number in any other number system in terms of digits ranging from 0 to 9 and then A – F only. This number system consists of only sixteen basic digits i.e. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F. Here 10 is represented as A, 11 as B and so on till 15 which is represented as F.

For Example: 47 in the Decimal number system can be represented as 2F in the Hexadecimal number system.

import java.io.\*;

class Dec2Hex

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader (new InputStreamReader(System.in));

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

int n=Integer.parseInt(br.readLine());

int r;

String s=""; //variable for storing the result

//array storing the digits (as characters) in a hexadecimal number system

char dig[]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};

while(n>0)

{

r=n%16; //finding remainder by dividing the number by 16

s=dig[r]+s; //adding the remainder to the result

n=n/16;

}

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

}

}

#### ****Question:****

Write a Program in Java to input a number in Decimal number system and convert it into its equivalent number in the Octal number system.

**Note:** Octal Number system is a number system which can represent a number in any other number system in terms of digits ranging from 0 to 7 only. This number system consists of only eight basic digits i.e. 0, 1, 2, 3, 4, 5, 6 and 7.

For Example: 25 in the Decimal number system can be represented as 31 in the Octal number system.

import java.io.\*;

class Dec2Oct

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader (new InputStreamReader(System.in));

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

int n=Integer.parseInt(br.readLine());

int r;

String s=""; //variable for storing the result

//array storing the digits (as characters) in the octal number system

char dig[]={'0','1','2','3','4','5','6','7'};

while(n>0)

{

r=n%8; //finding remainder by dividing the number by 8

s=dig[r]+s; //adding the remainder to the result and reversing at the same time

n=n/8;

}

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

}

}

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### ****Question:****

Write a Program in Java to input a number in Decimal number system and convert it into its equivalent number in the Binary number system.

**Note:** Binary Number system is a number system which can represent a number in any other number system in terms of 0 and 1 only. This number system consists of only two basic digits i.e. 0 and 1.

For Example: 25 in the Decimal number system can be represented as 11001 in the Binary number system.

import java.io.\*;

class Dec2Bin

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader (new InputStreamReader(System.in));

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

int n=Integer.parseInt(br.readLine());

int r;

String s=""; //variable for storing the result

char dig[]={'0','1'}; //array storing the digits (as characters) in a binary number system

while(n>0)

{

r=n%2; //finding remainder by dividing the number by 2

s=dig[r]+s; //adding the remainder to the result and reversing at the same time

n=n/2;

}

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

}

}

#### ****Question:****

Write a Program in Java to input a number and check whether it is a **Duck Number** or not.

**Note:** A Duck number is a number which has zeroes present in it, but there should be no zero present in the beginning of the number. For example 3210, 7056, 8430709 are all duck numbers whereas 08237, 04309 are not.

import java.io.\*;

class Duck\_No

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter any number : ");

String n=br.readLine(); //inputting the number and storing it in a String

int l=n.length(); //finding the length (number of digit) of the number

int c=0; //variable for counting number of zero digits

char ch;

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

{

ch=n.charAt(i); //extracting each digit and checking whether it is a '0' or not

if(ch=='0')

c++;

}

char f=n.charAt(0); //taking out the first digit of the inputted number

if(c>0 && f!='0')

System.out.println("It is a duck number");

else

System.out.println("It is not a duck number");

}

}

Write a program that encodes a word into Piglatin. To translate a word into a Piglatin word, convert the word into uppercase and then place the first vowel of the original word as the start of the new word along with the remaining alphabets. The alphabets present before the vowel being shifted towards the end followed by “AY”.  
Sample Input (1) : London, Sample Output (1) : ONDONLAY  
Sample Input (2) : Olympics, Sample Output (2) : OLYMPICSAY

import java.io.\*;

class Piglatin

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader (new InputStreamReader(System.in));

System.out.print("Enter any word: ");

String s=br.readLine();

s=s.toUpperCase(); //converting the word into Uppercase

int l=s.length();

int pos=-1;

char ch;

for(int i=0; i<l; i++)

{

ch=s.charAt(i);

if(ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')

{

pos=i; //storing the index of the first vowel

break;

}

}

if(pos!=-1) //printing piglatin only if vowel exists

{

String a=s.substring(pos); //extracting all alphabets in the word beginning from the 1st vowel

String b=s.substring(0,pos); //extracting the alphabets present before the first vowel

String pig=a+b+"AY"; //adding "AY" at the end of the extracted words after joining them

System.out.println("The Piglatin of the word = "+pig);

}

else

System.out.println("No vowel, hence piglatin not possible");

}

}

This is the Java programming code written in BlueJ which swaps the values of two Strings without using any third (temp) variable.  
  
import java.io.\*;

class Swap\_Strings

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader (new InputStreamReader (System.in));

System.out.print("Enter the 1st String : ");

String s1=br.readLine();

int len1=s1.length();

System.out.print("Enter the 2nd String : ");

String s2=br.readLine();

System.out.println("-------------------------------");

System.out.println("Strings Before Swapping : ");

System.out.println("1st String = "+s1);

System.out.println("2nd String = "+s2);

/\*Swapping Process Begins\*/

s1=s1+s2;

s2=s1.substring(0,len1);

s1=s1.substring(len1);

/\*Swapping Process Ends\*/

System.out.println("-------------------------------");

System.out.println("Strings After Swapping : ");

System.out.println("1st String = "+s1);

System.out.println("2nd String = "+s2);

}

}

#### Question:

Write a program to find the shortest and the longest word in a sentence and print them along with their length.

**Sample Input:** I am learning Java  
**Sample Output:**  
Shortest word = I  
Length = 1  
Longest word = learning  
Length = 8

import java.io.\*;

class Short\_long\_word

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter any sentence : "); //inputting the sentence

String s=br.readLine();

s=s+" "; //adding a space at the end, to extract the last word also

int len=s.length(); //finding the length of the sentence

String x="",maxw="",minw="";

char ch;

int p,maxl=0,minl=len;

for(int i=0;i<len;i++)

{

ch=s.charAt(i); //extracting characters of the string one at a time

if(ch!=' ')

{

x=x+ch; //adding characters to form word if character is not space

}

else

{

p=x.length();

if(p<minl) //checking for minimum length

{

minl=p;

minw=x;

}

if(p>maxl) //checking for maximum length

{

maxl=p;

maxw=x;

}

x=""; //emptying the temporary variable to store next word

}

}

System.out.println("Shortest word = "+minw+"nLength = "+minl);

System.out.println("Longest word = "+maxw+"nLength = "+maxl);

}

}

#### Question:

A bank intends to design a program to display the denomination of an input amount, up to 5 digits. The available denomination with the bank are of rupees 1000 , 500 , 100 , 50 , 20 , 10 , 5 , 2 , and 1.

Design a program to accept the amount from the user and display the break-up in descending order of denomination. (i.e. preference should be given to the highest denomination available) along with the total number of notes. [Note: Only the denomination used, should be displayed].

**Example:**

**INPUT**:   14788

**OUTPUT**:

DENOMINATIONS:

1000     x     14    =   14000  
500       x     1      =   500  
100       x     3      =   300  
50         x     1      =   50  
20         x     1      =   20  
10         x     1      =   10  
5           x     1      =   5  
2           x     1      =   2  
1           x     1      =   1  
————————————–  
TOTAL                =   14788  
————————————–  
Total Number of Notes = 23

[**Note:** This question came in ISC 2010 Practical Examination with a little addition of printing the amount in words according to the digits. Like, for 14788, it should print One Four Seven Eight Eight.]

import java.io.\*;

class Denominations

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int den[]={1000,500,100,50,20,10,5,2,1}; //storing all the denominations in an array

System.out.print("Enter any Amount: "); //Entering an amount

int amount=Integer.parseInt(br.readLine());

int copy=amount; //Making a copy of the amount

int totalNotes=0,count=0;

System.out.println("\nDENOMINATIONS: \n");

for(int i=0;i<9;i++) //Since there are 9 different types of notes, hence we check for each note.

{

count=amount/den[i]; // counting number of den[i] notes

if(count!=0) //printing that denomination if the count is not zero

{

System.out.println(den[i]+"\tx\t"+count+"\t= "+den[i]\*count);

}

totalNotes=totalNotes+count; //finding the total number of notes

amount=amount%den[i]; //finding the remaining amount whose denomination is to be found

}

System.out.println("--------------------------------");

System.out.println("TOTAL\t\t\t= "+copy); //printing the total amount

System.out.println("--------------------------------");

System.out.println("Total Number of Notes\t= "+totalNotes); //printing the total number of notes

}

}

Note: “\t” is used for giving “Tab spaces” and “\n” is used for going to the new line.

#### Question:

Write a program in JAVA to find the Prime factors of a number.

Prime factors of a number are those factors which are prime in nature and by which the number itself is completely divisible (1 will not be taken as prime number).

Few such numbers are:  
Prime Factors of 24 are 2, 2, 2, 3  
Prime Factors of 6 are 2, 3

import java.io.\*;

class PrimeFactors

{

public static void main(String args[]) throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int n;

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

n=Integer.parseInt(br.readLine());

System.out.print("The Prime Factors of "+n+" are : ");

int i=2;

while(n>1)

{

if(n%i == 0)

{

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

n=n/i;

}

else

i++;

}

}

}

[**Note:** The number entered should be within the Range 1-3999]

#### Question:

Write a program in Java to find the Roman equivalent of any Decimal number entered by the user. [The number entered should be within the Range 1-3999]

#### Brief Note on Roman Numerals:

The Roman numerals follow this basic pattern,  
1000 = M, 900 = CM, 500 = D, 400 = CD, 100 = C, 90 = XC, 50 = L, 40 = XL, 10 = X, 9 = IX, 5 = V, 4 = IV, 1 = I

The symbols “I”, “X”, “C”, and “M” can be repeated three times in succession, but no more. i.e. 234 can be represented as CCXXXIV, but 244 cannot be written as CCIV [Since we cannot repeat X more than 3 times successively].

(They may only appear more than three times if they appear non-sequentially, such as XXXIX.) “D”, “L”, and “V” can never be repeated.

import java.io.\*;

class Dec2Roman\_Method1

{

public static void main(String args[]) throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

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

int num=Integer.parseInt(br.readLine()); //accepting decimal number

if(num>0 && num<4000) //checking whether the number entered is within the range [1-3999]

{

/\*Saving the Roman equivalent of the thousand, hundred, ten and units place of a decimal number\*/

String thou[]={"","M","MM","MMM"};

String hund[]={"","C","CC","CCC","CD","D","DC","DCC","DCCC","CM"};

String ten[]={"","X","XX","XXX","XL","L","LX","LXX","LXXX","XC"};

String unit[]={"","I","II","III","IV","V","VI","VII","VIII","IX"};

/\*Finding the digits in the thousand, hundred, ten and units place\*/

int th=num/1000;

int h=(num/100)%10;

int t=(num/10)%10;

int u=num%10;

/\*Displaying equivalent roman number\*/

System.out.println("Roman Equivalent= "+thou[th]+hund[h]+ten[t]+unit[u]);

}

/\*Displaying an error message if the number entered is out of range\*/

else

System.out.println("nYou entered a number out of Range.nPlease enter a number in the range [1-3999]");

}

}

#### Question:

Write a program in Java to find the Least Common Multiple (L.C.M.) of two numbers entered by the user.

import java.io.\*;

class LcmMethod\_1

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int a,b,lcm=1;

System.out.print("Enter the 1st number : ");

a=Integer.parseInt(br.readLine());

System.out.print("Enter the 2nd number : ");

b=Integer.parseInt(br.readLine());

for(int i=a;i<a\*b;i++) //Even if you start the for loop by 1, you will get the answer, but starting it from either the first or the second number reduces the number of times the for loop is executed.

{

if(i%a==0 && i%b==0) //Checking the first number which is divisible by both the numbers

{

lcm=i;

break; //exiting from the loop, as we don’t need anymore checking after getting the LCM

}

}

System.out.println("L.C.M. = "+lcm);

}

}

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### Question:

Write a program to accept a sentence and print only the first letter of each word of the sentence in capital letters separated by a full stop.  
**Example :**  
INPUT SENTENCE : “This is a cat”  
OUTPUT : T.I.A.C.

import java.io.\*;

class Initials

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String s;

char x;

int l;

System.out.print("Enter any sentence: ");

s=br.readLine();

s=" "+s; //adding a space infront of the inputted sentence or a name

s=s.toUpperCase(); //converting the sentence into Upper Case (Capital Letters)

l=s.length(); //finding the length of the sentence</span>

System.out.print("Output = ");

for(int i=0;i<l;i++)

{

x=s.charAt(i); //taking out one character at a time from the sentence

if(x==' ') //if the character is a space, printing the next Character along with a fullstop

System.out.print(s.charAt(i+1)+".");

}

}

}

**This material contains some of the most important Programs for ICSE. The programs have been solved & cross checked.**