Practical Solutions

Semester 1

1. **Write a program to input a natural number less than 1000 and display it in words. Test your program on the sample data and some random data.**

   Sample input and output of the program

   Input: 29

   Output: TWENTY NINE

   Input: 17001

   Output: OUT OF RANGE

   Input: 119

   Output: ONE HUNDRED AND NINETEEN

   Input: 500

   Output: FIVE HUNDRED

```
import java.util.*;
class Number_Name
{
    public static void main(String Args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a number : ");
        int number = input.nextInt();
        if(number > 0 && number < 1000)
        {
            String ones [] = {"", "ONE", "TWO", "THREE", "FOUR", "FIVE", "SIX", "SEVEN", "EIGHT'
            String teens []= {"TEN", "ELEVEN", "TWELVE", "THIRTEEN", "FOURTEEN", "FIFTEEN", "SIX
            String tens []= {"", "TEN", "TWENTY", "THIRTY", "FORTY", "FIFTY", "SIXTY", "SEVENTY'
            String name = "";
            if (number >= 100)
            {
                name += ones[number / 100] + " HUNDRED ";
                number %= 100;
            }
            if (number >= 20)
            {
                name += tens[number / 10] + " ";
                number %= 10;
            }
            else if (number >= 10 && number <= 19)
            {
                name += teens[number - 10] + " ";
                number = 0;
            }
            if (number > 0 )
            {
                name += ones[number] + " ";
            }
            System.out.println(name.trim());
        }
        else if ( number == 0 )
        {
            System.out.println("ZERO");
        }
        else
        {
            System.out.println("OUT OF RANGE");
        }
    }
}
```

2. Encryption is a technique of encoding messages to maintain their secrecy. A String array of size 'n' where 'n' is greater than 1 and less than 10, stores single sentences (each sentence ends with a full stop) in each row of the array.

   Write a program to accept the size of the array. Display an appropriate message if the size is not satisfying the given condition. Define a string array of the inputted size and fill it with sentences row-wise. Change the sentence of the odd rows with an encryption of two character ahead of the original character. Also change the sentence of the even rows by storing the sentence in reverse order. Display the encrypted sentences as per the sample data given below.

   Test your program on the sample data and some random data

   Input: n=4
   IT IS CLOUDY.
   IT MAY RAIN.
   THE WEATHER IS FINE.
   IT IS COOL.
   Output:
   KV KU ENQWFA.
   RAIN MAY IT.
   VJG YGCVJGT KU HKPG.
   COOL IS IT.
   Input: n=13
   Output: INVALID ENTRY

```java
import java.util.*;
class Encryption
{
    public static void main(String Args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the size of the array : ");
        int n = input.nextInt();
        input.nextLine();
        if( n > 0 && n < 10)
        {
            String ar[] = new String[n];
            for(int i = 0; i < n; i++)
            {
                System.out.print("Enter the "+i+" sentence : ");
                ar[i] = input.nextLine().toUpperCase();
                if(! ar[i].endsWith("."))
                {
                    System.out.println("The sentence does not end in a full stop.");
                    System.out.print("Enter the "+i+" sentence : ");
                    ar[i] = input.nextLine().toUpperCase();
                }
            }
            for(int i = 0; i < n; i++)
            {
                if ((i%2) == 1)
                {
                    String word [] = (ar[i].substring(0, ar[i].length() - 1)).split(" ");
                    String reversed = "";
                    for(int j = word.length-1; j >= 0; j--)
                    {
                        reversed += word [j];
                        if( j != 0)
                        {
                            reversed += " ";
                        }
                    }
                    reversed += ".";
                    ar[i] = reversed;
                }
                else
                {
                    String encrypted = "";
                    for(int j = 0; j < ar[i].length(); j++)
                    {
                        char current = ar[i].charAt(j);
                        if(Character.isLetter(current))
                        {
                            char incremented = (char) (current + 2);
                            if (incremented > 'Z')
                            incremented = (char) ('A' + (incremented - 'Z' - 1));
                            encrypted += incremented;
                        }
                        else
                        {
                            encrypted +=current;
                        }
                    }
                    ar[i] = encrypted;
                }
            }
            for(int i = 0; i < n; i++)
            {
                System.out.println(ar[i]);
            }
        }
        else
        {
            System.out.println("INVALID ENTRY");
        }
    }
}
```

3. **Design a program which accepts your date of birth in dd mm yyyy format. Check whether the date entered is valid or not. If it is valid, display "VALID DATE", also compute and display the day number of the year for the date of birth. If it is invalid, display "INVALID DATE" and then terminate the program.**

   Testing of the program

   Input: Enter your date of birth in dd mm yyyy format

   05

   01

   2010

   Output: VALID DATE

   5

Input: Enter your date of birth in dd mm yyyy format

01

07

2023

Output: VALID DATE

Input: Enter your date of birth in dd mm yyyy format

34

06

2010

Output: INVALID DATE

```
import java.util.*;
class ValidDate
{
    public static void main(String Args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter your date of birth in dd mm yyyy format");
        System.out.print("dd : ");
        int day = input.nextInt();
        System.out.print("mm : ");
        int month = input.nextInt();
        System.out.print("yyyy : ");
        int year = input.nextInt();
        int maxDays [] = {31,28,31,30,31,30,31,31,30,31,30,31};
        int day_number = 0;
        if((year%4) == 0)
        {
            maxDays[1] += 1;
        }
        if(month > 0 && month <= 12)
        {
            if(day > 0 && day <= maxDays[month-1])
            {
                System.out.println("VALID DATE");
                for(int i = 0; i < month; i++)
                {
                    day_number += maxDays[i];
                }
                day_number += day;
                System.out.println(day_number);
            }
            else
            {
                System.out.println("INVALID DATE");
            }
        }
        else
        {
            System.out.println("INVALID DATE");
        }
    }
}
```

Semester 2

1. **Design a program which accepts your date of birth in dd mm yyyy format. Check whether the date entered is valid or not. If it is valid, display "VALID DATE", also compute and display the day number of the year for the date of birth. If it is invalid, display INVALID DATE" and then terminate the program.**

```
import java.util.*;
class SM2_Q1
{
    public static void main(String Args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter your birth date in dd mm yyyy format");
        System.out.print("dd : ");
        int dd = input.nextInt();
        System.out.print("mm : ");
        int mm = input.nextInt();
        System.out.print("yyyy : ");
        int yyyy = input.nextInt();
        int days []= {31,28,31,30,31,30,31,31,30,31,30,31};
        if((yyyy%4)==0)
        {
            days[1] = 29;
        }
        if(mm>=1 && mm<=12)
        {
            if(dd<=days[mm-1])
            {
                System.out.println("VALID DATE");
                int day_count = 0;
                for(int i=0;i<=(mm-2);i++)
                {
                    day_count+=days[i];
                }
                day_count+=dd;
                System.out.println("Number of days : "+day_count);
            }
            else
            {
                System.out.println("INVALID DATE");
            }
        }
        else
        {
            System.out.println("INVALID DATE");
        }
    }
}
```

2. **Write a program to declare a square matrix A[][] of order MxM where 'M' is the number of rows and the number of columns, such that M must be greater than 2 and less than 10. Accept the value of M as user input.**

   Display an appropriate message for an invalid input. Allow the user to input integers into this matrix.

   Perform the following tasks:

   a) Display the original matrix.

   b) Check if the given matrix is symmetric or not. A square matrix is said to be symmetric, if the element in the ith row and jth column is equal to the element of the jth row and ith column.

   c) Find the sum of the elements of left diagonal and the sum of the elements of right diagonal of the matrix and display them.

```
import java.util.*;
class Q2_2S
{
    public static void main(String Args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the value of 'M': ");
        int M = input.nextInt();
        if(M>2 && M<10)
        {
            int A[][]= new int[M][M];
            int i,j;
            for(i=0;i<M;i++)
            {
                for(j=0;j<M;j++)
                {
                    System.out.print("["+i+"] ["+j+"] : ");
                    A[i][j]=input.nextInt();
                }
            }.
            System.out.println("ORIGINAL MATRIX");
            for(i=0;i<M;i++)
            {
                for(j=0;j<M;j++)
                {
                    System.out.print(A[i][j]+" ");
                }
                System.out.println();
            }
            boolean isSymmetric = false;
            for(i=0;i<M;i++)
            {
                for(j=0;j<M;j++)
                {
                    if(A[i][j]==A[j][i]);
                    {
                        isSymmetric = true;
                    }
                }
            }
            if(isSymmetric)
            {
                System.out.println("THE GIVEN MATRIX IS SYMMETRIC.");
            }
            else
            {
                System.out.println("THE GIVEN MATRIX NOT IS SYMMETRIC.");
            }
            int sum_left=0;
            int sum_right=0;
            for(i=0;i<M;i++)
            {
                sum_left+=A[i][i];
                sum_right+=A[i][(M-1)-i];
            }
            System.out.println("The sum of the left diagonal = "+sum_left);
            System.out.println("The sum of the right diagonal = "+sum_right);
        }
        else
        {
            System.out.print("The value of M should be between 2 and 10");
        }
    }
}
```

3. **A number N will be a monocot number if the sum of its extreme (first and last) digits is equal to the sum of mean(all digits except first and last) digits. If the sum of extreme digits is not equal to the sum of mean digits, the number is called dicot number.**

```
import java.util.*;
class SM2_Q3
{
    public static void main(String Args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a number : ");
        int N = input.nextInt();
        String number = Integer.toString(N);.
        int extreme = number.charAt(0) -'0' + number.charAt(number.length()-1) -'0';
        String mean_digits = number.substring(1, number.length() - 1);
        int mean = 0;
        for(int i =0;i<mean_digits.length();i++)
        {
            mean += Character.getNumericValue(mean_digits.charAt(i));
        }
        System.out.println("Sum of mean digits : "+mean);
        System.out.println("Sum of extreme digits : "+extreme);
        if(mean==extreme)
        {
            System.out.println(N+" is a monocot number.");
        }
        else
        {
            System.out.println(N+" is a dicot number.");
        }
    }
}
```