Hard programs

1. Write a Java Program to Convert a Given Number of Days in Terms of Years, Weeks & Days.

    Sample Input & Output:

```
Enter the number of days: 756

        No. of years: 2

        No. of weeks: 3

        No. of days: 5
```

Test cases:

1. 38
2. 3.6
3. 0
4. -365
5. -45

```
import java.util.*;
class Daysconverter
{
    public static void main(String args[])
    {
        try
        {
            int years=0,weeks=0,days=0;
            int rem1=0,rem2=0;
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter the  number of days:");
            days=sc.nextInt();
            if(days>0)
            {
                years=days/365;
                rem1=days%365;
                weeks=rem1/7;
                rem2=rem1%7;
                System.out.println("No. of Years:"+years);
                System.out.println("No. of weeks:"+weeks);
                System.out.println("No. of Days:"+rem2);
            }
            else
            {
                System.out.println("Please Enter positive value only!");
            }

        }
    catch (Exception e)
    {
      System.out.println("Please Enter integer value only!");
    }

  }


}
```

Given a date, return the corresponding day of the week for that date.

The input is given as three integers representing the day, month and year respectively.

Return the answer as one of the following values {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}.

Example 1:

Input: day = 31, month = 8, year = 2019

Output: "Saturday"

Example 2:

Input: day = 18, month = 7, year = 1999

Output: "Sunday"

Example 3:

Input: day = 15, month = 8, year = 1993

Output: "Sunday"

Constraints:

· The given dates are valid dates between the years 1971 and 2100.

```
import java.time.DayOfWeek;
import java.time.LocalDate;
import java.util.Scanner;

class DayOfWeekFinder
{
  public static void main(String[] args)
  {


      Scanner sc = new Scanner(System.in);

      System.out.println("Enter the day:");
      int day = sc.nextInt();
```

```
            System.out.println("Enter the month:");
            int month = sc.nextInt();

            System.out.println("Enter the year:");
            int year = sc.nextInt();
            if(year>1971 && year<2100)
            {
                LocalDate date = LocalDate.of(year, month, day);
                DayOfWeek dayOfWeek = date.getDayOfWeek();
                System.out.println("The day of the week is: " + dayOfWeek);
            }
            else
            {
                System.out.println("Invalid date. Please enter a valid date.");
            }
    }

}
```

2. Write a program to find the number of student users in the college, get the total users, staff users details from the client. Note for every 3 staff user there is one Non teaching staff user assigned by default.

Sample Input:

Total Users: 856

Staff Users: 126

Sample Output:

Student Users: 688

Test Cases:

1. Total User: 0

2. Total User: -143

3. Total User: 1026, Staff User: 1026

4. Total User: 450, Staff User: 540

5. Total User: 600, Staff User: 450

```
import java.util.Scanner;

class Users
{
    public static void main(String[] args)
    {

        int Total_users=0,staff_users=0,nt_users=0,student_users=0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Total Users:");
        Total_users = sc.nextInt();
        System.out.println("Enter the Staff Users:");
        staff_users = sc.nextInt();
        if(Total_users==staff_users || Total_users<staff_users ||Total_users<=0 ||staff_users<=0)
        {
            System.out.println("Invalid input");
        }
        else
        {
            nt_users=staff_users/3;
            student_users=Total_users-(staff_users+nt_users);
            System.out.println("Student Users are:"+student_users);
        }


    }

}
```

3. Write a program to print the number of factors and to print the nth factor of the given number.

Sample Input:

Given Number: 100

N = 4

Sample Output:

Number of factors = 9

4th factor of 100 = 5

Test Cases:

1. Given Number = 512 , N = 6
2. Given Number = 343 , N = 7
3. Given Number = 1024 , N = 0
4. Given Number = -6561 , N = 3
5. Given Number = 0 , N = 2

```
import java.util.Scanner;

class Factors {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number: ");
        int num = scanner.nextInt();

        System.out.print("Enter the value of N: ");
        int n = scanner.nextInt();

        if (n <= 0) {
            System.out.println("N should be a positive integer.");
            return;
        }
        int[] factors = new int[100];
        int count = 0;
        for (int i = 1; i <= num; i++) {
            if (num % i == 0) {
                factors[count] = i;
                count++;
            }
        }

        System.out.println("Number of factors = " + count);

        if (n <= count) {
            System.out.println(n + "th factor of " + num + " = " + factors[n - 1]);
        } else {
            System.out.println("N is greater than the number of factors.");
        }
    }
}
```

4. Write a program to print n prime numbers after nth Prime number

Sample Input:

N = 3

Sample Output:

3rd Prime number is 5

3 prime numbers after 5 are: 7, 11, 13

Test cases:

1. N = P

2. N = 0

3. N = -4

4. N = 11

5. N = 7.2

```
import java.util.*;
import java.util.ArrayList;
class primenum {
    public static void main(String[] args) {
        int n, i, j,flag = 0;
        ArrayList<Integer> list = new ArrayList<>();
        Scanner s = new Scanner(System.in);
        System.out.println("enter the number:");
        n = s.nextInt();
        for (i = 2; i < 1000; i++) {
            flag = 1;

            for (j = 2; j <= i / 2; ++j) {
                if (i % j == 0) {
                    flag = 0;
                    break;
                }
            }
            if (flag == 1)
                list.add(i);
        }
        System.out.print("\n"+n+"th prime number is:"+list.get(n-1));
        for(i=0;i<n;i++)
            System.out.print("\n"+list.get(n+i));

    }
}
```

5. Write a Program to create a list of all numbers in a range which are perfect squares and the sum of the digits of the number is less than 10.

Sample Input & Output:

```
Enter lower range: 1

Enter upper range: 40

        [1, 4, 9, 16, 25, 36]
```

Test case:

1. Enter lower range: 50

   Enter upper range: 100

2. Enter lower range: 5

   Enter upper range: 8

3. Enter lower range: 10

   Enter upper range: 5

4. Enter lower range: 500

Enter upper range: 500

5. Enter lower range: 0

Enter upper range: -100

```
import java.util.Scanner;

public class PerfectSquaresInRange {
    public static void main(String[] args) {
        int l, u, c = 0;
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter lower range:");
        l = sc.nextInt();
        System.out.println("Enter upper range:");
        u = sc.nextInt();

        int[] arr = new int[100];

        if (l < 0 || l == u || l > u || u<0)
        {
            System.out.println("Invalid input");
        }
        else
        {
            for (int i = l; i <= u; i++) {
                double sqrt = Math.sqrt(i);
                if (sqrt == Math.floor(sqrt)) {
                    int rem, sum = 0;
                    int temp = i;

                    while (temp > 0) {
                        rem = temp % 10;
                        sum += rem;
                        temp = temp / 10;
                    }

                    if (sum < 10) {
                        arr[c] = i;
                        c++;
                    }
                }
            }

            if (c > 0)
            {
                System.out.println("Perfect squares with sum of digits less than 10:");
                for (int i = 0; i < c; i++)
                {
                    System.out.print(arr[i] + " ");
                }
            }
            else
            {
                System.out.println("No perfect squares between " + l + " and " + u);
            }
        }

    }
}
```

6. Write a program to print unique permutations of a given number

Sample Input:

Given Number: 143

Sample Output:

Permutations are:

134

143

314

341

413

431

Test cases:

1. 0
2. 111
3. 505
4. -143
5. -598

```
public class h6 {
    public static void main(String[] args)
    {
        String str = "143";
        int n = str.length();
        h6 permutation = new h6();
        permutation.permute(str, 0, n - 1);
    }

    private void permute(String str, int l, int r)
    {
        if (l == r)
            System.out.println(str);
        else {
            for (int i = l; i <= r; i++) {
                str = swap(str, l, i);
                permute(str, l + 1, r);
                str = swap(str, l, i);
            }
        }
    }

    public String swap(String a, int i, int j)
    {
        char temp;
        char[] charArray = a.toCharArray();
        temp = charArray[i];
        charArray[i] = charArray[j];
        charArray[j] = temp;
        return String.valueOf(charArray);
    }
}
```

7. Write a Program to create an array with the First Element as the Number and Second Element as the Square of the Number.

Sample Input:

Enter the lower range:45

Enter the upper range:49

Sample Output:

```
[(45, 2025), (46, 2116), (47, 2209), (48, 2304), (49, 2401)]
```

Test case:

1. Enter lower range: 50

   Enter upper range: 100

2. Enter lower range: 5

   Enter upper range: 8

3. Enter lower range: 10

Enter upper range: 5

4. Enter lower range: 500

   Enter upper range: 500

5. Enter lower range: 0

   Enter upper range: -100

```
import java.util.Scanner;

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

        System.out.print("Enter the lower range: ");
        int l = scanner.nextInt();

        System.out.print("Enter the upper range: ");
        int u = scanner.nextInt();

        if (l <= u)
        {

            int size = u - l + 1;
            int[][] numsq = new int[size][2];

            for (int i = 0; i < size; i++)
            {
                int num = l + i;
                numsq[i][0] = num;
                numsq[i][1] = num * num;
            }

            System.out.print("[");
            for (int i = 0; i < size; i++)
            {
                System.out.print("(" + numsq[i][0] + ", " + numsq[i][1] + ")");
                if (i < size - 1) {
                    System.out.print(", ");
                }
            }
            System.out.println("]");
        }
        else
        {
            System.out.println("Invalid input: Lower range should be less than or equal to upper range.");
        }
    }
}
```