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

```
Scanner input=new Scanner(System.in);
int num=input.nextInt();
int years=num/365;
System.out.println("years: "+years);
int weeks=(num%365)/7;
System.out.println("weeks: "+weeks);
int days=(num%365)%7;
System.out.println("Days: "+days);
```

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.

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

```
Scanner input=new Scanner(System.in);
System.out.print("Total Users: ");
int total_user=input.nextInt();
System.out.print("Staff Users: ");
int staff_user=input.nextInt();
int non_tech=staff_user/3;
int student_user=total_user-(staff_user+non_tech);
System.out.println("Student Users: "+student_user);
```

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

Sample Input:
Given Number: 100
N = 4
Sample Output:
Number of factors = 9
4th factor of 100 = 5

```
Scanner input=new Scanner(System.in);
int num=input.nextInt();
int n=input.nextInt();
int a[]=new int[100];
int x=0;
for(int i=1;i<=num;i++)
{
    if(num%i==0) {
        a[x] = i;
        x++;
    }
}
System.out.println("Number of factors = "+x);
System.out.println(n+" factor of "+num+" = "+a[n-1]);
```

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

```
Scanner input=new Scanner(System.in);
int n=input.nextInt();
int a[]=new int[100];
int x=0;
for(int i=2;i<=100;i++)
{
    int fact=0;
    for(int j=1;j<=i;j++)
    {
        if(i%j==0)
            fact++;
    }
    if(fact==2) {
        a[x] = i;
        x++;
    }
}
System.out.println(n+" prime number is "+a[n-1]);
System.out.print(n+" prime numbers after "+a[n-1]+" are: ");
for(int i=n;i<n+n;i++)
{
    System.out.print(a[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]

```
import java.util.Scanner;
public class ak {
    public static void main(String[] args)
    {
        Scanner input=new Scanner(System.in);
        int lower=input.nextInt();
        int upper=input.nextInt();
        int i=0,x=1;
        while(i<upper)
        {
            int sum=0;
            int y=x*x;
            int t=y;
            while(y!=0)
            {
                int rem=y%10;
                sum=sum+rem;
                y=y/10;
            }
            if(sum<10)
            {
                System.out.print(t+" ");
            }
            i=x*x;
            x++;
        }
    }
}
```

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

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

```
import java.util.Scanner;
public class ak
{
    public static void print(int a[])
    {
        for(int i=0;i<a.length;i++)
        {
            System.out.print(a[i]+" ");
        }
        System.out.println();
    }
    public static void swap(int a[],int i,int j)
    {
        int temp=a[i];
        a[i]=a[j];
        a[j]=temp;
    }
    public static void per(int a[],int t)
    {
        if(t==a.length)
        {
            print(a);
            return;
        }
        for(int i=t;i<a.length;i++)
        {
            swap(a,i,t);
            per(a,t+1);
            swap(a,i,t);
        }
    }
    public static void main(String[] args)
    {
        Scanner input=new Scanner(System.in);
        int a[]={1,4,3};
        per(a,0);
    }
}
```

Test cases:

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

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)]

```
Scanner input=new Scanner(System.in);
int lower=input.nextInt();
int upper= input.nextInt();
for(int i=lower;i<=upper;i++)
{
    System.out.println("("+i+","+(i*i)+")"+" ");
}
```

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

8. Develop a JAVA code to display the balance. Include the following members:

- Design a class to represent a bank account.
- **Data Members:** Name of the depositor, Account number, Type of account(Savings/Current), Balance amount in the account(Minimum balance is Rs.500.00)
- **Methods:**
  1. To read account number, Depositor name, Type of account.
  2. To deposit an amount (Deposited amount should be added with it)
  3. To withdraw an amount after checking balance(Minimum balance must be Rs.500.00

  Note : Assume that balance amount = 10000

Test Cases

1. 100, Raja, S, 8000
2. Raja, 100, S, 9000
3. 101, Rani, S, 12000
4. 102, Ragu, W, 8000
5. 103, Ravi, C, 10000

```
6. import java.util.Scanner;
   class Bank_Account
   {
        String name,type;
        int acc_num;
        double bal;
        Bank_Account(String n,int a,String t,double b)
        {
            name=n;
            acc_num=a;
            type=t;
            bal=b;
        }
        void deposit(double d)
        {
            if(d>0)
                bal=bal+d;
            else
                System.out.println("Invalid amt");
        }
        void withdraw(double w)
        {
            if(w>0 && w<=bal)
                bal=bal-w;
            else
                System.out.println("Invalid amt");
        }
        void display()
        {
            System.out.println("Name: "+name);
            System.out.println("balance: "+bal);
        }

    }
    public class ak
    {
        public static void main(String[] args)
        {
            Scanner input=new Scanner(System.in);
            String n,t;
            int a;
            double b;
            System.out.print("Enter the name: ");
            n=input.next();
            System.out.print("Enter Account number: ");
            a=input.nextInt();
            System.out.print("Enter the type of account: ");
            t= input.next();
            System.out.print("Enter the balance amt: ");
            b=input.nextInt();
            Bank_Account bank=new Bank_Account(n,a,t,b);
            bank.display();


        }
    }
```

9. Develop a code to Reverse and Add a Number until you get a Palindrome.

Sample Input   If 7325 is input number, then

7325 (Input Number) + 5237 (Reverse Of Input Number) = 12562

12562 + 26521 = 39083

39083 + 38093 = 77176

77176 + 67177 = 144353

144353 + 353441 = 497794 (Palindome)

```
import java.util.Scanner;
public class ak
{
    public static int revnum(int num)
    {
        int rev=0;
        while(num!=0)
```

```java
        {
            int rem=num%10;
            rev=rev*10+rem;
            num=num/10;
        }
        return rev;
    }
    public static boolean check(int num)
    {
        int rev= revnum(num);
        if(num==rev)
            return true;
        else
            return false;
    }
    public static void add(int num)
    {
        if(check(num)) {
            System.out.println("palindrome");
        }
        else
        {
            while(!check(num))
            {
                int rev=revnum(num);
                int sum=num+rev;
                System.out.println(num+"+"+rev+"="+sum);
                num=sum;
            }
        }
    }

    public static void main(String[] args)
    {
        Scanner input=new Scanner(System.in);
        int num=input.nextInt();
        add(num);
    }
}
```

Test Cases

1. 8765
2. -8765
3. 0
4. EIGHT FIVE
5. 87.57

10. Create Customer class with deposit() and withdraw() as synchronized methods. Declare AccountNo, AccName and Balance as Instance Variables inside the class. From the main class, Input the amount for withdraw() operation and if requested amount is not available in existing Balance amount, withdraw() method should be temporarily suspended using wait() method until deposit() method receives the input for amount, to be added in the existing Balance amount and then withdraw() would be completed in a successful manner. Develop the above scenario using Synchronization and Inter-Thread Communication.

Note : existing Bank balance amount 10000
Sample Input : 12000, 3000
Sample Output : Withdraw operation success, balance amount 1000

Test Cases

1. 11000, 4000
2. -10000, -2000
3. 0, 0
4. EIGHT SEVEN, FIVE
5. 100.67, 200.68

11. Given an integer n, return a string array answer (1-indexed) where:
answer[i] == "FizzBuzz" if i is divisible by 3 and 5.
answer[i] == "Fizz" if i is divisible by 3.
answer[i] == "Buzz" if i is divisible by 5.
answer[i] == i (as a string) if none of the above conditions are true.

Example 1:
Input: n = 3
Output: ["1","2","Fizz"]

```
int n;
Scanner sc=new Scanner(System.in);
System.out.println("enter the value of n: ");
n= sc.nextInt();
for(int i= 1;i<=n;i++){
    if(i%3==0 && i%5==0){
        System.out.println("FIZZ BUSS");
    }
    else if(i%3==0){
        System.out.println("FIZZ");
    }
    else if(i%5==0){
        System.out.println("BUSS");
    }
    else {
        System.out.println(i);
    }
}
```

Test Case

Test Case	Inputs
1.	n = 5
2.	n = 10
3.	n = 12
4.	n = 18
5.	n = 20

12. Write a Java program to find the common elements in two array of Positive integer
Sample Input:

[1, 2, 3, 4]
[2, 4, 5, 6, 7]
Expected output: [2, 4]

```
int a[]=new int[]{1,2,3,4};
int b[]=new int[]{2,4,2,6,7};
int len=b.length;
for(int i=0;i<len;i++)
{
    for(int j=i+1;j<len;j++)
    {
        if(b[i]==b[j])
        {
            for(int k=j;k<len-1;k++)
            {
                b[k]=b[k+1];
            }
            j--;
            len--;
        }
    }
}
for(int i=0;i<a.length;i++)
{
    for(int j=0;j<len;j++)
    {
        if(a[i]==b[j]) {
            System.out.print(a[i]);
        }
    }
}
```

Test Case

Test Case	**Inputs-1**	**Inputs-2**
1.	[1, 2, 3, 4]	[4,5,6,7,8]
2.	[a, b, c, d]	[a, b, c, d]
3.	[1, -2, 3, 4]	[1,-2,5,7,8]
4.	[@, #, 34, 45]	[@,#,%,$,]
5.	[45,78,56,89]	[92,34,56,-78,-90]

13. **Given a string** `s` **consisting of words and spaces, return the length of the** last **word** ***in the string.*** **A** word **is a maximal substring consisting of non-space characters only. There will be at least one word, consists of only English letters and spaces ' '.**
    Example 1:
    Input: s = "Hello World"
    Output: 5
    Explanation: The last word is "World" with length 5.

```
Scanner sc=new Scanner(System.in);
int wl=0;
System.out.println("enter the string: ");
String s1=sc.nextLine();
String w[]=s1.split(" ");
if(w.length>0){
    wl=w[w.length-1].length();
}
else{
    wl=0;
}
System.out.println("length is: "+ wl);
```

Test Case

Test Case	**Inputs-1**
1.	Maximal Substring Consisting
2.	**lea@st one wor2d**
3.	1254 98076
4.	& * ( ) % # $
5.	letters and spaces

14. Write a program to read a character until a ***** is encountered. Also count the number of uppercase, lowercase, and numbers entered by the users.

```
Sample Input:
Enter * to exit…
Enter any character: W
Enter any character: d
Enter any character: A
Enter any character: G
Enter any character: g
Enter any character: H
Enter any character: *
Sample Output:
Total count of lower case:2
Total count of upper case:4
Total count of numbers =0
```

Test Case

Test Case	**Inputs-1**
1.	1,7,6,9,5
2.	S, Q, l, K,7, j, M
3.	M, j, L, &, @, G
4.	D, K, I, 6, L, *
5.	*, K, A, e, 1, 8, %, *

15. Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

Symbol	**Value**
I	**1**
V	**5**
X	**10**
L	**50**
C	**100**
D	**500**
M	**1000**

For example, 2 is written as II in Roman numeral, just two ones added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.
Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

- **I can be placed before V (5) and X (10) to make 4 and 9.**
- **X can be placed before L (50) and C (100) to make 40 and 90.**
- **C can be placed before D (500) and M (1000) to make 400 and 900.**
  Given a roman numeral, convert it to an integer.

Example:
Input: s = "III"
Output: 3

```java
import java.util.Scanner;
public class ak
{
    public static int value(char c)
    {
        if(c=='I')
            return 1;
        if(c=='V')
            return 5;
        if(c=='X')
            return 10;
        if(c=='L')
            return 50;
        if(c=='C')
            return 100;
        if(c=='D')
            return 500;
        if(c=='M')
            return 1000;
        return -1;
    }
    public static int romantodecimal(String str)
    {
        int res=0;
        for(int i=0;i<str.length();i++)
        {
            int s1=value(str.charAt(i));
            if(i+1<str.length())
            {
                int s2=value(str.charAt(i+1));
                if(s1>=s2)
                    res=res+s1;
                else
                {
                    res=res+s2-s1;
                    i++;
                }
            }
            else
                res=res+s1;
        }
        return res;
    }
    public static void main(String[] args)
    {
        Scanner input=new Scanner(System.in);
        String str=input.next();
        System.out.println(romantodecimal(str));
    }
}
```

Test Case	Inputs
1.	LVIII
2.	MCMXCI
3.	V
4.	LZAII
5.	MCCDTIV

16. **Given two strings ransomNote and magazine, return true if ransomNote can be constructed by using the letters from magazine and false otherwise. Each letter in magazine can only be used once in ransomNote.**

Example 1:
Input: ransomNote = "a", magazine = "b"
Output: false

```
import java.util.Scanner;
public class ak
{
    public static void main(String[] args)
    {
        Scanner input=new Scanner(System.in);
        System.out.print("RansomeNOte: ");
        String ransomenote=input.next();
        System.out.print("Magzine: ");
        String magazine=input.next();
        int t1[]=new int[ransomenote.length()];
        int t2[]=new int[magazine.length()];
        for(int i=0;i<ransomenote.length();i++)
        {
            t1[i]=ransomenote.charAt(i);
        }
        for(int i=0;i<magazine.length();i++)
        {
            t2[i]=magazine.charAt(i);
        }
        int len=ransomenote.length();
        int c=0;
        for(int i=0;i<ransomenote.length();i++)
        {
            for(int j=0;j<magazine.length();j++)
            {
                if(t1[i]==t2[j]) {
                    c++;
                    break;
                }
            }
        }
        if(len==c)
            System.out.println("true");
        else
            System.out.println("false");
    }
}
```

Test Case

Test Case	Inputs
1.	ransomNote = "aa", magazine = "ab"
2.	ransomNote = "aa", magazine = "aab"
3.	ransomNote = "abc", magazine = "abc"
4.	ransomNote = "good", magazine = "better"
5.	ransomNote = "xyz", magazine = "123"

17. **You are given an m x n binary matrix mat of 1's (representing soldiers) and 0's (representing civilians). The soldiers are positioned in front of the civilians. That is, all the 1's will appear to the left of all the 0's in each row.**
A row i is weaker than a row j if one of the following is true:
The number of soldiers in row i is less than the number of soldiers in row j.
Both rows have the same number of soldiers and i < j. Return the indices of the k weakest rows in the matrix ordered from weakest to strongest.

Example 1:
Input: mat =
[[1,1,0,0,0],
[1,1,1,1,0],
[1,0,0,0,0],
[1,1,0,0,0],
[1,1,1,1,1]],
k = 3
Output: [2,0,3]
Explanation:
The number of soldiers in each row is:
- Row 0: 2
- Row 1: 4
- Row 2: 1
- Row 3: 2
- Row 4: 5
The rows ordered from weakest to strongest are [2,0,3,1,4].

```
import java.util.*;
class ak{
    public static void main(String[] args){
        Scanner a=new Scanner(System.in);
        System.out.println("enter no of rows");
        int row=a.nextInt();
        System.out.println("enter no of col");
        int col=a.nextInt();
        int arr[][]=new int[row][col];
        int arr1[]=new int[row];
        int arr2[]=new int[row];
        for(int i=0;i<row;i++){
            int c=0;
            for(int j=0;j<col;j++){
                arr[i][j]=a.nextInt();
                if(arr[i][j]==1){
                    c++;
                }
            }
            arr1[i]=c;
            arr2[i]=c;
        }
        Arrays.sort(arr2);
        System.out.println("no of least elemnt index you want");
        int index=a.nextInt();
        List<Integer> li=new ArrayList<>();
        for(int i:arr2){
            for(int j=0;j<row;j++){
                if(i==arr1[j]){
                    li.add(j);
                }
            }
        }
        for(int i=0;i<index;i++){
            System.out.print(li.get(i)+" ");
        }
    }
}
```

Example 2:
Input: mat =
[[1, 0, 0, 0],
 [1, 1, 1,1],
 [1, 0, 0, 0],
 [1, 0, 0,0]],
 k = 2
Output: [0,2]
Explanation:
The number of soldiers in each row is:
- Row 0: 1
- Row 1: 4
- Row 2: 1
- Row 3: 1
The rows ordered from weakest to strongest are [0, 2, 3, 1].

18. Given an integer num, return the number of steps to reduce it to zero. In one step, if the current number is even, you have to divide it by 2, otherwise, you have to subtract 1 from it.

Example 1:
Input: num = 14
Output: 6
Explanation:
Step 1) 14 is even; divide by 2 and obtain 7.
Step 2) 7 is odd; subtract 1 and obtain 6.

Step 3) 6 is even; divide by 2 and obtain 3.
Step 4) 3 is odd; subtract 1 and obtain 2.
Step 5) 2 is even; divide by 2 and obtain 1.
Step 6) 1 is odd; subtract 1 and obtain 0.

```
Scanner sc=new Scanner(System.in);
System.out.println("enter a number: ");
int n=sc.nextInt();
int s=0;
while(n>0){
    if(n%2==0){
        n=n/2;

    }
    else{
        n=n-1;

    }
    s=s+1;
}
System.out.println(s);
```

Test Case

Test Case	**Inputs**
1.	n = 5
2.	n = 10
3.	n = 12
4.	n = 18
5.	n = 20

19. Develop a programme that uses Multiple Inheritance concepts to compute a student's grades in six subjects. The total and aggregate are then calculated, and the student's grade is displayed. If the student achieves an aggregate of more than 75%, the grade is Distinction. If the aggregate is between 60 and 75, the grade is First Division. If the aggregate is between 50 and 60, the grade is Second Division. If the aggregate is between 40 and 50, the grade is Third Division. Otherwise, the grade is FAIL.

Sample Input & Output:

```
Enter the marks in python: 90
Enter the marks in c programming: 91
Enter the marks in Mathematics: 92
Enter the marks in Physics: 93
Enter the marks in Chemistry: 92
Enter the marks in Professional Ethics: 93
Total= 551
Aggregate = 91.83
Class: DISTINCTION
```

Test Case

Test Case	**Inputs**
1.	18, 76,93,65,63,98
2.	73,78,79,75,87,0
3.	98,106,120,95,98,34

4.	96,73, -85,95,84,98
5.	78,59.8,76,79,97,67

20. Write a program to calculate tax given the following conditions:

a. If income is less than or equal to 2,50,000 then no tax

b. If taxable income is 2,50,001 – 5,00,000 the charge 10% tax

c. If taxable income is 5,00,001 – 10,00,000 the charge 20% tax

d. If taxable income is above 10,00,001 then charge 30% tax

Sample Input:

Enter the income: 600000

Sample Output:

Taxable Income: 350000

Tax= 35000

Test Case

Test Case	**Inputs**
1.	400700
2.	2789239
3.	150000
4.	00000
5.	-125486