**1.BOAT TRAVEL**

You're travelling in a boat, the water currents sometimes favours you and sometimes they slow you down. You need to calculate the time to reach the destination knowing the boat speed and the water current speed.   
A boat moves at a constant speed of x kmph without water flow. The water flow also affects the speed boat.  
Given the ideal boat speed and the water flow speed, write a program to find the time required to reach a given distance.  
  
**Input format:**  
First Line of the input contains a single integer value that denotes the ideal boat speed in kmph.  
Second Line of the input contains a single integer value that denotes the water flow speed in kmph.(+ve when it favours the boat and -ve when it disfavours the boat)  
Third Line of the input contains a single integer value that denotes the distance to be covered in km.  
  
**Output format:**  
A single Integer value that denotes the time taken to reach the given distance.  
  
**Assume**: The time taken will always be an integer.  
  
**Sample input 1:**  
20  
5  
100  
**Sample Output** 1:  
4  
  
**Sample input 2:**  
25  
-5  
100  
**Sample output 2:**  
5

**ANSWER:**

import java.util.Scanner;

public class BoatTravel {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int idealspeed=input.nextInt();

int waterflowspeed=input.nextInt();

int distance=input.nextInt();

int timetaken=(distance/(idealspeed+waterflowspeed));

System.out.println(timetaken);

}

}

**2.BIGGER NUMBER**

Suriya loves playing with numbers. He comes up with  a new game idea of creating a greater number than the given number by changing exactly one digit. Now he has to find the number of ways in which this is possible. Help Suriya to develop a program to find the number of ways to obtain a bigger number.  
  
**Input format:**  
Input contains a single integer that denotes the given number.  
  
**Output format:**  
The output contains a single integer that denotes the number of ways.  
  
**Sample Input 1:**  
10  
**Sample Output 1:**  
17  
  
**Sample Input 2:**  
32310  
**Sample output 2:**  
36   
  
**Explanation:**  
First digit 3 can be changed into 4,5,6,7,8,9 so **6** possibilities  
Second digit 2 can be changed into 3,4,5,6,7,8,9 so **7** possibilities  
Third digit 3 can be changed into 4,5,6,7,8,9 so **6** possibilities  
Fourth digit 1 can be changed into 2,3,4,5,6,7,8,9 so **8** possibilities  
Fifth digit 0 can be changed into 1,2,3,4,5,6,7,8,9 so **9** possibilities  
**Total**  
6+7+6+8+9 = 36

**ANSWER:**

import java.util.Scanner;

public class BiggerBumberSuriya {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int N=input.nextInt();

String str=String.valueOf(N);

int num=0,sum=0;

for(int i=0;i<str.length();i++) {

num=Character.

getNumericValue

(str.charAt(i));

int count=0;

for(int j=num+1;j<=9;j++) {

count++;

}

sum+=count;

}

System.out.println(sum);

}

}

**3.GREATEST COMMON PRIME DIVISOR**

Sapna has learnt about prime numbers and Greatest common divisor. Now she experiments by finding the greatest common prime divisor.  
Help her develop a program by combining her knowledge in both the concepts to find the greatest common prime divisor(gcpd).  
Given two numbers, write a program to find the greatest common prime divisor.  
  
**Note :**  
    If there is no such numbers, print -1.  
  
**Input Format:**  
First Line of the input contains a single integer that denotes the first number.  
Second Line of the input contains a single integer that denotes the second number.  
  
**Output Format:**   
The output contains a single integer that denotes the gcpd of the two inputs  
  
**Sample Input 1:**  
12  
18  
**Sample Output 1:**  
3  
  
**Sample Input 2:**  
12  
13  
**Sample Output 2:**  
-1

**ANSWER:**

import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;

public class GreatestCommonPrimeDivisor {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int n=input.nextInt();

int m=input.nextInt();

int gcd=0;

ArrayList<Integer> gcds=new ArrayList<Integer>();

ArrayList<Integer> primes=new ArrayList<Integer>();

for(int i=2;i<=n&&i<=m;i++) {

if(n%i==0&&m%i==0) {

gcd=i;

gcds.add(gcd);

}

}

for(Integer temp:gcds) {

boolean isPrime=true;

for(int i=2;i<temp;i++) {

if(temp%i==0) {

isPrime=false;

}

}

if(isPrime) {

primes.add(temp);

}

}

Collections.sort(primes);

if(primes.isEmpty()) {

System.out.println(-1);

}

else {

System.out.println

(primes.get(primes.size()-1));

}

}

}

**4.POWER PAIRS**

Your friend challenges you to play a game, in which you have to find the number of Power pairs in a series of numbers. A Power pair is formed when a number-b is divisible by another number-a that occurs before the number-b in the given sequence. You can easily win the challenge by developing a program to find it.  
  
**Input Format**  
First Line of the input contains a single integer that denotes the size of the array - n.  
Second line consists of n space separated integers that denote the array values.  
  
**Output Format**  
The output consists a single integer that denotes the number of Power pairs in the given sequence.  
  
**Sample Input 1:**  
3  
1 3 2  
**Sample Output 1:**  
2  
  
**Explanation :**  
For sequence = [1, 3, 2]  
The sorted pairs are: (1, 3), (1, 2).      
So output should be 2.  
  
**Sample Input 2:**  
3  
2 4 8  
**Sample Output 2:**  
3

**ANSWER:**

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import java.util.Scanner;

import java.util.Set;

import java.util.TreeSet;

public class PowerPairs {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int n=input.nextInt();

int a[]=new int[n];

for (int i = 0; i < a.length; i++) {

a[i]=input.nextInt();

}

int count=0;

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

for(int i=j+1;i<n;i++) {

if(a[i]%a[j]==0) {

count++;

}

}

System.out.println(count);

}

}

**5.DIAGONAL DIFFERENCE**

Roshan just found out that matrix can be implemented using 2d arrays. He is good in programming. So he tries to develop  a program to do his math homework. Help Roshan to write a program to do his math homework.  
In his math home work, a square matrix of size N is given. He needs to calculate the absolute difference between the sums of its diagonals.  
Help Roshan by writing a program.  
  
**Input Format:**  
The first line contains a single integer denotes the size of the matrix, N.  
The next N  lines denote the matrix's rows, with each line containing N space-separated integers describing the columns.  
  
**Output Format:**  
Print the absolute difference between the two sums of the matrix's diagonals as a single integer.  
  
**Sample Input 1:**  
3  
11 2 4  
4 5 6  
10 8 -12  
**Sample Output 1:**  
15  
  
**Explanation:**  
The primary diagonal elements are:  
11 5 -12  
Sum across the primary diagonal: 11 + 5 - 12 = 4  
  
The secondary diagonal elements are:  
4 5 10  
Sum across the secondary diagonal: 4 + 5 + 10 = 19  
Absolute Difference = |4 - 19| = **15**.

**ANSWER:**

import java.util.ArrayList;

import java.util.Scanner;

public class DiagnalDifference {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int N=input.nextInt();

int[][] a=new int[N][N];

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

a[i][j]=input.nextInt();

}

}

int i=0;

int firstsum=0,secondsum=0;

for(int j=0;j<N;j++) {

firstsum+=a[i][j];

i++;

}

int n=0;

for(int j=N-1;j>=0;j--) {

secondsum+=a[n][j];

n++;

}

System.out.println

(Math.abs(firstsum-secondsum));

}

}

**6.EXCHANGE POSITIONS IN ARRAY**

Nikitha is very interested in programming with arrays, she tries to exhange the positions of the elements by swapping the element and the number at the place which it occupies.  
For Example,  
Consider the array elements.  
1 3 4 2  
The element 1 in given array , is at position 1, so 1 is placed at position 1 in output array  
The element 3 in given array , is at position 2, so 2 is placed at position 3 in output array  
The element 4 in given array , is at position 3, so 3 is placed at position 4 in output array   
The element 2 in given array , is at position 4, so 4 is placed at position 2 in output array  
The output array becomes 1,4,2,3  
  
Develop a program to help Nikitha.  
  
**Input Format:**  
First Line of the input contains a single integer that denotes the size of the array n  
Second  line consists of n space separated integer array values.  The input array values will always be a permutation of numbers from 1 to n.  
  
**Output Format:**  
The output consists of n space separated integers that denote the output array values.  
  
**Sample Input 1:**  
4  
1 3 4 2  
**Sample Output 1:**  
1 4 2 3  
  
**Sample Input 2:**  
3  
1 2 3  
**Sample Output 2:**  
1 2 3

**ANSWER:**

import java.util.Scanner;

public class ExchangePositionsInArray {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int N=input.nextInt();

int a[]=new int[N];

int b[]=new int[N];

for (int i = 0; i < a.length; i++) {

a[i]=input.nextInt();

}

for (int i = 0; i < a.length; i++) {

if((i+1)==a[i]) {

b[i]=i+1;

}

else {

b[a[i]-1]=i+1;

}

}

for(int i=0;i<b.length;i++) {

System.out.print(b[i]+" ");

}

}

}