Question:

Given two strings, a and b , that may or may not be of the same length, determine the minimum number of character deletions required to make a and b anagrams. Any characters can be deleted from either of the strings.

Input :

test cases,t

two strings a and b, for each test case

Output:

Desired O/p

Constraints :

string lengths<=10000

Program:

import java.util.\*;

class TestClass {

public static void main(String args[] ) throws Exception {

int out = 0;

Scanner r = new Scanner(System.in);

int t = r.nextInt();

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

{

String a = r.next();

String b = r.next();

if((a.length()>=1)&&(b.length()>=1)&&(a.length()<=10000)&&(b.length()<=10000))

{

int c1[] = new int[26];

int c2[] = new int[26];

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

{

c1[a.charAt(j)-'a']++;

}

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

{

c2[b.charAt(j)-'a']++;

}

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

{

out += Math.abs(c1[j]-c2[j]);

}

System.out.println(out);

out = 0;

}

}

}

}

Question:

Akash and Vishal are quite fond of travelling. They mostly travel by railways. They were travelling in a train one day and they got interested in the seating arrangement of their compartment. The compartment looked something like

So they got interested to know the seat number facing them and the seat type facing them. The seats are denoted as follows :

Window Seat : WS

Middle Seat : MS

Aisle Seat : AS

You will be given a seat number, find out the seat number facing you and the seat type, i.e. WS, MS or AS.

INPUT

First line of input will consist of a single integer T denoting number of test-cases. Each test-case consists of a single integer N denoting the seat-number.

OUTPUT

For each test case, print the facing seat-number and the seat-type, separated by a single space in a new line.

Program:

import java.util.\*;

class TestClass {

public static void main(String args[] ) throws Exception {

Scanner r = new Scanner(System.in);

int t = r.nextInt();

if(t>=1&&t<=100000)

{

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

{

int n=r.nextInt();

if(n>=1&&n<=108)

{

int ch = n%12;

switch(ch)

{

case 0:System.out.println((n-11)+" WS");

break;

case 1:System.out.println((n+11)+" WS");

break;

case 2:System.out.println((n+9)+" MS");

break;

case 3:System.out.println((n+7)+" AS");

break;

case 4:System.out.println((n+5)+" AS");

break;

case 5:System.out.println((n+3)+" MS");

break;

case 6:System.out.println((n+1)+" WS");

break;

case 7:System.out.println((n-1)+" WS");

break;

case 8:System.out.println((n-3)+" MS");

break;

case 9:System.out.println((n-5)+" AS");

break;

case 10:System.out.println((n-7)+" AS");

break;

case 11:System.out.println((n-9)+" MS");

break;

default:System.out.print("");

break;

}

}

}

}

}

}

Question:

You have been given a String S. You need to find and print whether this string is a palindrome or not. If yes, print "YES" (without quotes), else print "NO" (without quotes).

Input Format

The first and only line of input contains the String S. The String shall consist of lowercase English alphabets only.

Output Format

Print the required answer on a single line.

Constraints 1=|S|=100

Note

String S consists of lowercase English Alphabets only.

Program:

import java.util.\*;

class palli {

static boolean isPallindrome(String strr)

{

int i=0;

int j=strr.length()-1;

while(i<j)

{

if(strr.charAt(i)!=strr.charAt(j))

{

return false;

}

i++;

j--;

}

return true;

}

public static void main(String args[] ){

Scanner s = new Scanner(System.in);

String str = s.nextLine();

if (isPallindrome(str))

{

System.out.println("YES");

}

else

{

System.out.println("NO");

}

}

}

Question:

Your program is to use the brute-force approach in order to find the Answer to Life, the Universe, and Everything. More precisely... rewrite small numbers from input to output. Stop processing input after reading in the number 42. All numbers at input are integers of one or two digits.

Program:

import java.util.\*;

import java.io.\*;

class TestClass {

public static void main(String args[] ) throws Exception {

Scanner sc = new Scanner(System.in);

int input = sc.nextInt();

if(input!=42)

{

System.out.println(input);

}

while(input!=42)

{

input = sc.nextInt();

if(input!=42)

{

System.out.println(input);

}

}

}

}

Question:

In our school days, all of us have enjoyed the Games period. Raghav loves to play cricket and is Captain of his team. He always wanted to win all cricket matches. But only one last Games period is left in school now. After that he will pass out from school.

So, this match is very important to him. He does not want to lose it. So he has done a lot of planning to make sure his teams wins. He is worried about only one opponent - Jatin, who is very good batsman.

Raghav has figured out 3 types of bowling techniques, that could be most beneficial for dismissing Jatin. He has given points to each of the 3 techniques.

You need to tell him which is the maximum point value, so that Raghav can select best technique.

3 numbers are given in input. Output the maximum of these numbers.

Program:

import java.util.\*;

import java.io.\*;

class TestClass {

public static void main(String args[] ) throws Exception {

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

int b = sc.nextInt();

int c = sc.nextInt();

if(a>b && a>c)

{

System.out.println(a);

}

else if(b>a && b>c)

{

System.out.println(b);

}

else

{

System.out.println(c);

}

}

}