Week End Assignment-1

Question 5

You are given two non-empty strings S and T of equal lengths. S contains the characters ‘0’, ‘1’ and ‘?’, whereas T contains ‘0’ and ‘1’ only. Your task is to convert S into T in minimum number of moves. In each move, you can do one of these changes:

* change a ‘0’ in S to ‘1’
* change a ‘?’ in S to ‘0’ or ‘1’
* swap any two characters in S

As an example, suppose S = “01??00” and T = “001010”. We can transform S into T in 3 moves:

* Initially S = “01??00”
* Move 1 – change S[2] to ‘1’.
* S becomes “011?00”
* Move 2 – change S[3] to ‘0’.
* S becomes “011000”
* Move 3 – swap S[1] with S[4].
* S becomes “001010” S is now equal to T.

**Input**

The first line of input is an integer C (C ≤ 200) that indicates the number of test cases. Each case consists of two lines. The first line is the string S consisting of ‘0’, ‘1’ and ‘?’. The second line is the string T consisting of ‘0’ and ‘1’. The lengths of the strings won’t be larger than 100.

**Output**

For each case, output the case number first followed by the minimum number of moves required to convert S into T. If the transition is impossible, output −1 instead. Check the output example for the exact format.

**Input Example**

3

01??00

001010

01

10

110001

000000

**Output Example**

Case 1: 3

Case 2: 1

Case 3: -1

PROGRAM:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace assign4in13

{

class Program

{

//Solve method which returns the moves required for convert S to T

static int Solve(int len, char[] S, char[] T)

{

//S has 4 type of difference from T

int[] type = new int[4];

//loop for each element in char array

for (int i = 0; i < len; ++i)

{

//0 in S replaced by 1 in T

if (S[i] == '0' && T[i] == '1')

type[0]++;

//1 in S replaced by 0 in T

if (S[i] == '1' && T[i] == '0')

type[1]++;

//? in S replaced by 1 in T

if (S[i] == '?' && T[i] == '1')

type[2]++;

//? in S replaced by 0 in T

if (S[i] == '?' && T[i] == '0')

type[3]++;

}

//Minimum moves between the type as answer

int ans = Math.Min(type[0], type[1]);

//when type0 is greater than 1

if (type[0] >= type[1])

{

//diff between type 1 and 0 becomes 0

type[0] -= ans;

//compare with 3 for ? and changing the ans with twice of diff

if (type[3] - type[0] >= 0)

{

ans += 2 \* type[0];

ans += type[2] + type[3] - type[0];

}

else

{

ans += 2 \* type[3];

ans += type[2] + type[0] - type[3];

}

}

//when type1 is greater than 0

else if (type[0] < type[1])

{

type[1] -= ans;

if (type[2] - type[1] >= 0)

{

ans += 2 \* type[1];

ans += type[3] + type[2] - type[1];

}

}

return ans;

}

static void Main(string[] args)

{

const int MAX = 255;

//p,q are string to get input converted to S[] and T[] after assignment

string p = "";

string q = "";

char[] S = new char[MAX];

char[] T = new char[MAX];

int tcase;

int a, b;

Console.WriteLine("Give the number of test cases for analysis");

tcase = Int32.Parse(Console.ReadLine());

Console.WriteLine("Enter the Strings S and T:");

//loop for number of test cases

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

{

a = 0;

b = 0;

p = Console.ReadLine();

q = Console.ReadLine();

S = p.ToCharArray();

T = q.ToCharArray();

int len = S.Length;

//loop for each char element in S for Comparison

for (int k = 0; k < len; ++k)

{

if (S[k] == '1')

a++;

if (T[k] == '1')

b++;

}

//returns -1 if conversion is not possible

if (a > b)

Console.WriteLine("Case {0} : {1}", t, -1);

//return the number of moves required for conversion

else

Console.WriteLine("Case {0} : {1}", t, Solve(len, S, T));

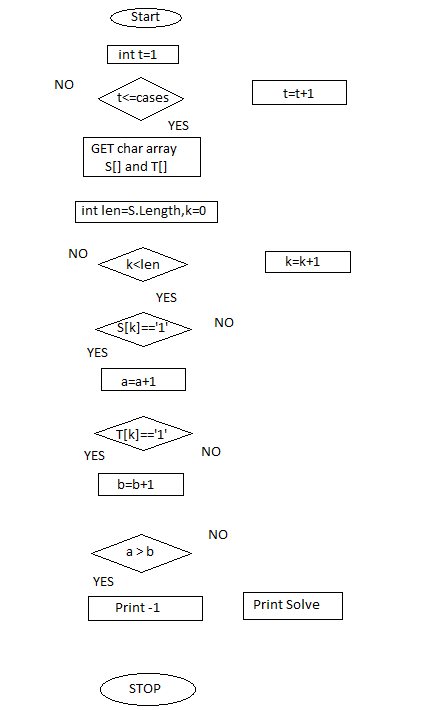
}

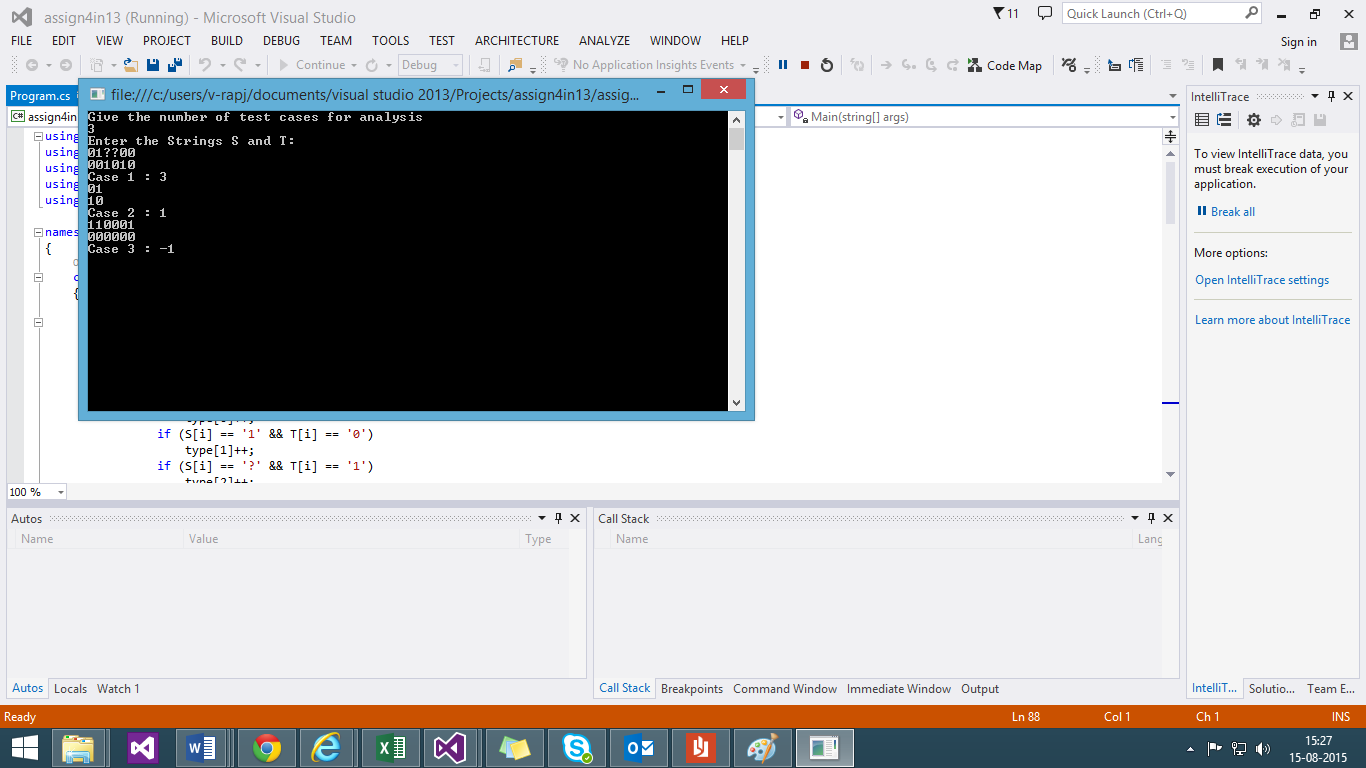
Console.ReadKey();

}

}

}





**Algorithm:**

1. The first two are main in rearranging and the rest two computes the number of 1 required to be inserted to duplicate the T string
2. The input from user decided the number of test cases which is done in a for loop.
3. The inner for loop is for the comparison of each element in the string of S and T to check it can be modified or not.
4. If it can be modified Solve function is called.
5. There are 4 cases of difference between T and S they are taken as types in Solve function
6. The difference in the cases are computed by comparison of the occurrence of each case and answer is changed depending on the minimum value.