Interview Code's

```
using System;
using System.Collections.Generic;

public class HelloWorld
{
    public static void SwitchForGood(string T, string P)
    {

      Console.WriteLine($"Working On {T} and {P} ");
      List<string> whyYou = new List<string>();

       whyYou.Add("Struggle");
       whyYou.Add("900/Month");
       whyYou.Add("Many Following");
       whyYou.Add("To give life for Manoj BHauji");
       whyYou.Add("To Gannya Educate");
       whyYou.Add("Don't Drink");
       whyYou.Add("Aai/Baba");

       foreach(var varx in whyYou){
       Console.WriteLine(varx );
       }
    }

    public static void Main(string[] args)
    {
        string S = "Temperament";
        string P = "Patience";
        SwitchForGood(S, P);
    }
}
```

```
class Bird
{
    public virtual void fly(){

        Console.WriteLine("Bird can fly");
    }
}
class Parrot : Bird
{
    public override void fly()
    {
        Console.WriteLine("Parrot can fly");
    }
}
class Program
{
    static void Main(string[] args)
    {
        Bird b = new Bird();
        b.fly();
        Bird b1 = new Parrot();
        b1.fly();
    }
}
Output:
Bird can fly
Parrot can fly
```

Interview Code's

```
I1
{
    void static print();
}
I2
{
    void print();
}
Class Implement: I1, I2
{
    public Implement(
      public i1 = I1;
      public i2 = I2;
    )
// prob1 -
    implementingMethod(){
          i1.print();
          i2.print();
    }
    // prob 2-
    I1 l = I1.print();
}

async methodMain(
    await p1.Method1();
    await p2.Method2();
    whenAll(p1.Method1,p2.Method2)
);
async Method1 (){
 await
}
```

```
// sum of evenn numbers in Linq
 var even_num = from num in context.numbers
                  where (numbers.number %2 == 0)
                   select num;

foreach(number in even_num){
}
```

```
// Top 2 salary by Linq

var highestSalary = (from emp in Employee
                     order by emp.salary Descending
                     select emp).Top(2)
```

```
//Checking Palindrom
string word_pal = "madam"
```

Interview Code's

```
char [] word_palArr = word_pal.ToCharArray(); //"m","a","d","a","m"
Array.Revers(word_palArr);
string reverseWordString = new string(word_palArr)

       if( (word_palArr =reverseWordString )){
        Console.WriteLine("This is Palindrom")
       }


--------------------------

///Deloitte
///Table 1 list empploye - Employee
// Table 2 Mapping Manager,Lead - EmployeeDetails
fetch list of employees reporting to "Selnior Manager"
get

var listEmp = from emp in Context.Employee
              Inner Join empD in Context.EmployeeDetails
              On emp.Id = empD.Id
              where empD.ReportM = "Selnior Manager"
              select new { emp.name};

foreach()


--------------------------

//output : Employee Id : A101

//please modify code
public class Employee
{
private string _id;
public Employee(string Id)
{

    _Id=Id
}
public void GetDetails(_Id){
    Console.Write("Employee Id :" this._Id);
}
//TO DO
var obj=new Employee("A101");
var txt=obj.ToString();
Console.Write("txt);

---------------------------


-- Linq
-- select first 3 charecter of Aditya
-- Table employee

select first3 = (from emp in context.employee
                 select substring(emp.fname,3)
                 select emp
                 )

foreach ( employee from  first3 )
{
}

-----------------------
```

Interview Code's

LINQ query to get the sum of the even numbers from the list:

```
Var sumOfEvenNumbers = from num in context.Number
                       where (numbers.number %2 == 0)
                       select num
```

To write the string as per row pass :

```
using System;
public class HelloWorld
{

    public static void ConvertMethod(string s, int numbRow)
    {
        // Calculate the length of each row
        int rowLength = (int)Math.Ceiling((double)s.Length / numbRow);
        // Initialize an array to hold each row
        string[] rows = new string[numbRow];
        // Fill the rows with characters from the string
        for (int i = 0; i < s.Length; i++)
        {
            int rowIndex = i / rowLength;
            if (rowIndex < numbRow)
            {
                rows[rowIndex] += s[i];
            }
        }
        // Print the rows
        foreach (var row in rows)
        {
            Console.WriteLine(row);
        }
    }

    public static void Main(string[] args)
    {
        string s = "NHKYFVHGTPMHI";
        int numbRow = 3;
        ConvertMethod(s, numbRow);
    }
}
```

How to assign num to List =

```
int[] num = { 1, 2, 3, 4 };
List<int> listNumbers = new List<int>();
listNumbers.AddRange(num);
```

arrayNumber = {1,2,3,4} I should get the multiplication other place number suppose at 1th place need multiplication of other like 2 * 3 * 4 = 24 at 2nd place 1 * 3* 4 = 12 oputput should be = {24 12 8 6}

```
int[] arrayNumber = { 1, 2, 3, 4 };
        int[] output = new int[arrayNumber.Length];
```

Interview Code's

```
        for (int i = 0; i < arrayNumber.Length; i++)
        {
            int product = 1;
            for (int j = 0; j < arrayNumber.Length; j++)
            {
                if (i != j)
                {
                    product *= arrayNumber[j];
                }
            }
            output[i] = product;
        }
```

SLB Client Interview -

To count the number of times two adjacent 1s appear in a given 2D array (either side by side in a row or up and down in a column), you can implement the following code in C#. Here's how you can do it:

```
using System;
class Program
{
    static void Main()
    {
        int[,] matrix =
        {
            { 1, 1, 0, 1 },
            { 1, 0, 1, 0 },
            { 1, 1, 0, 0 }
        };
        int count = CountAdjacentOnes(matrix);
        Console.WriteLine($"Number of adjacent 1s: {count}");
    }
    static int CountAdjacentOnes(int[,] matrix)
    {
        int rows = matrix.GetLength(0); // gives number of rows = 3
        int cols = matrix.GetLength(1); // gives number of columns = 3
        int count = 0;
        // Check for horizontal adjacent 1s
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols - 1; j++)
            {
                if (matrix[i, j] == 1 && matrix[i, j + 1] == 1)
                {
                    count++;
                }
            }
        }
        // Check for vertical adjacent 1s
        for (int i = 0; i < rows - 1; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                if (matrix[i, j] == 1 && matrix[i + 1, j] == 1)
                {
                    count++;
                }
            }
```

Interview Code's

```
        }
        return count;
    }
}
```

Ascendion -

```
Input => wwffwfeeaf
Output => w3f4e2a1

string inputFinder = "wwffwfeeaf"
public FindOccurence(string inputFinder){

  char [] InputArr = Array.ToChar(inputFinder);
"w","w","f","f","w","f","e","e","a","f"
  List<string> outputFinder = new List<string> outputFinder{};
  public int counter =  0;
   for(int i=0 ; i <= InputArr.Length ; i++){
         for(int j=i+1 ; j <= InputArr.Length ; j++)
            if(InputArr[i] == InputArr[j] ){
                outputFinder.Add(nputArr[i]);
                counter ++;
                outputFinder.Add(counter);

            }else{
                counter = 1;
            }

            return outputFinder;
            counter = 0;
   }
   string Output =  Array.Join(outputFinder);
   return Output;
}
```

```
public class A
{
     static A(){
       Console.WriteLine ("Static ");
      }
      private A(){
       Console.WriteLine ("private ");
     }
     public A(int num){
       Console.WriteLine ("public ");
    }
}
public class B
{
    public static void Main(string[] args)
    {
       A obj= new A();
       Console.WriteLine ("Main");
    }
}
```

```
- We cant call Private Constructor
```

Interview Code's

```
- Static will run first and then Public
-  We should pass argument in Public Constructor
  A obj= new A(123);
```

So Output will be
Static
Public
Main

---

give me sort of array number input without using Array.Sort in C#

```
class Program
{
    static void Main()
    {
        int[] numbers = { 5, 3, 8, 4, 2, 7, 1, 6 };

        // Bubble Sort
        for (int i = 0; i < numbers.Length - 1; i++)
        {
            for (int j = 0; j < numbers.Length - 1 - i; j++)
            {
                if (numbers[j] > numbers[j + 1])
                {
                    // Swap the elements
                    int temp = numbers[j];
                    numbers[j] = numbers[j + 1];
                    numbers[j + 1] = temp;
                }
            }
        }
        // Output the sorted array
        Console.WriteLine("Sorted array:");
        foreach (int num in numbers)
        {
            Console.Write(num + " ");
        }
    }
}
```
