```
List<int> list = new List<int>()
{ 1, 2, 3, 4,5,6,7,8,9,10};

List<string> list2 = new List<string>()
{
   "ahmed mohamdey","mohamed ahmed","mo salah","sayd ahmed"
};

var Names = from l in list
            where l > 5
            select l;

var Names = list2.Where(x => x.Contains("ahmed")).ToList();

//foreach(var name in Names)
//{
//    Console.WriteLine($"Name : {name}");
//}

list2.Remove("mohamed ahmed");
list2.AddRange(new string[] { "saeed saleh", "ahmed morad", "ahmed ahmed" });

foreach (var name in Names)
{
   Console.WriteLine($"Name : {name}");
}
```

هذه اذا كان الشرط على عمود من ضمن الاعمده المراد عرضها

```
var customers = GetData.GetCustomers().Select(x => new
                                {
                                   custName = x.name,
                                   custPhone = x.telephone,
                                   custage = x.age
                                }).Where(x => x.custage > 30);
```

هذه اذا كان الشرط على عمود ليس من ضمن الاعمده المراد عرضها

```
var customers = GetData.GetCustomers().Where(x => x.age > 30).Select(x => new
                                {
                                   custName = x.name,
                                   custPhone = x.telephone,
                                });
```

طريقة اخرى لاستحدام الوير مع تمرير بارامبتر انتيجر زيادة

```
var custo = GetData.GetCustomers().Where((x, i) => x.age > 30 && i == 1).Select(x => new
                        {
                           x.name,
                           x.telephone
                        });
```

وهذه بإستخدام الكويري او محاكاه لها

```
var customers = from cust in GetData.GetCustomers()
            where cust.age > 30
            select new
            {
               custName = cust.name,
               custPhone = cust.telephone
            };
```

طريقه اخرى باستخدام الكويري والوير بعد الكويري

```
var customers = from cust in GetData.GetCustomers()
            select new
            {
               cust.name,
               cust.telephone,
               cust.age
            } into c
            where c.age > 30
            select c;


foreach (var customer in customers)
{
   Console.WriteLine(customer.name);
}
```

هذه لترتيب العناصر تصاعديا بناء على عمود واحد

```
var orderList = GetData.GetCustomers().OrderBy(x => x.age);
```

هذه لترتيب العناصر تنازليا بناء على عمود واحد

```
var orderList = GetData.GetCustomers().OrderByDescending(x => x.age);
```

هذه لترتيب العناصر تنازليا بناء على العمر وتصاعديا بناء على عمود الاسماء وهي طريقة ترتيب العناصر على اكثر من عمود

```
var orderList = GetData.GetCustomers().OrderByDescending(x => x.age).ThenBy(x =>
x.name);
```

هذه لترتيب العناصر بناء على العمر تصاعديا وبناء على الاسماء تنازليا

```
var orderList = GetData.GetCustomers().OrderBy(x => x.age).ThenByDescending(x =>
x.name);
```

هذه لترتيب العناصر باستخدام الكويري وبعد العنصر اذا ما حددت الترتيب تلقائي بيكون تصاعدي

```
var orderList = from x in GetData.GetCustomers()
            orderby x.age descending, x.name ascending
            select x;
```

هذه لعرض اول صف من القائمة لكن اذا كانت القائمة فارغة سيعطيك خطأ في الرن تايم

```
var FirstList = GetData.GetCustomers().First();
```

هذه نفس السابق لاكن تقبل نل او قائمة فارغه وبامكانك ان تتحقق من قيمتها بأف شرطية

```
var FirstList = GetData.GetCustomers().FirstOrDefault();
```

هذه نفس السابق بس تجيب لك اول صف من الاشخاص الي اعمارهم فوق الثلاثين

```
var FirstList = GetData.GetCustomers().FirstOrDefault(x => x.age > 30);
```

هذه نفس السابقة بالضبط ولاكن قمت باعطائها قيمه مبدئية او ديفولت في حال كان الناتج لا شي او نل

```
var FirstList = GetData.GetOrders().FirstOrDefault(x => x.id > 1, new Order { id = 20, total = 30
});
```

هذه نفس جميع م سبق لاكن باستخدام الكويري

```
var FirstList = (from x in GetData.GetCustomers()
             orderby x.age
             select x).First();
```

وهذه تقوم بنفس الدالة السابقة بس تجيب اخر صف وتنطبق عليها جميع العمليات السابقة

```
var LastList = GetData.GetCustomers().Last();

Console.WriteLine(FirstList.name);

foreach (var order in orderList)
{
   Console.WriteLine("Name :{0} : Age :{1}", order.name, order.age);
```

هذه الدالة للحصول على الكائن من خلال الاندكس واذا كان الاندكس اكبر من عدد القائمة بيعطيك خطأ

```
var element = GetData.GetCustomers().ElementAt(0);
```

هذه نستخدمها فقط في حال انها ارجعت قيمة نل ونشيك عليها ولا نقدر نمرر لها ديفولت فاليو

```
var element = GetData.GetCustomers().ElementAtOrDefault(0);
```

هذه الداله ترجع فقط قيمه وحده وتعطيك خطا اذا كانت القائمة عددها اكبر من واحد
او اذا كان عددها صفر وبامكانك تمرر لها شرط يكون القيمة المرجعه منه قيمة واحده فقط
وبامكانك لا تمرر لها شرط

```
var single = GetData.GetCustomers().Single(x => x.id == 102);
Console.WriteLine(single.name);
```

وهذه الدالة تستخدم لعد عناصر القائمة اذ لم تستخدم لها شرط ستعد جميع العناصر
واذا مررت لها شرط سوف تعد عناصر القائمة بناء على الشرط

```
var count = GetData.GetCustomers().Count(x => x.age > 30);
```

هذه الدالة تقوم لك بارجاع اكبر قيمة في العمود المختار بين القوسين

```
var max = GetData.GetCustomers().Max(x => x.age);
Console.WriteLine(max);
```

وههذه الدالة تقوم بارجاع الكائن كامل بناء على اكبر قيمة في العمود المختار

```
var max = GetData.GetCustomers().MaxBy(x => x.age);
Console.WriteLine(max.name, max.age);
```

هذه الدالة تقوم بحساب المتوسط بناء على العمود المختار

```
var average = GetData.GetCustomers().Average(x => x.spendAverage);
Console.WriteLine(average);
```

هذه الدالة تقوم بحساب المجموع بناء على العمود المختار

```
var sum = GetData.GetCustomers().Sum(x => x.age);
Console.WriteLine(sum);
```

هذه الدالة نستخدمها لربط جدولين بينهم علاقة ون تو ماني
نقوم اولا بكتابة الجدول الذي يحتوي على علاقة الون ومن ثم نقوم بكتابة اسم الدالة
وداخل القوسين نكتب اولا اسم الجدول الذي يحتوي على علاقة الماني
ومن ثم نكتب العمود الذي يمثل البرايمري كيي ومن ثم بعده نكتب العمود الذي يمثل الفورين كيي
ومن ثم نكتب القيمة النهائية او المرجعه من هذا الربط

```
var cats = GetData.GetCategoryies();
var custs = GetData.GetCustomers();

var result = cats.Join(custs, cats => cats.id, custs => custs.categoryid,
  (cats, custs) => new
  {
    FullName = custs.name,
```

```
        CatogoryName = cats.name
    });

  foreach (var res in result)
  {
    Console.WriteLine(res.FullName + " : " + res.CatogoryName);
  }
```

هذه نفس السابقة بس باستخدام الكويري

```
var result = from cat in cats
        join cust in custs

        on cat.id equals cust.categoryid
        select new
        {
          FullName = cust.name,
          CatogoryName = cat.name
        };

  foreach (var res in result)
  {
    Console.WriteLine(res.FullName + " : " + res.CatogoryName);
  }
```

هذه الداله نستخدمها عندما نريد عمل ربط من ناحية اليسار بمعنى ان ناتي بجميع اعمدة الجدول الي ع اليسار مع ما م يتقاطع معها من جدول اليمين

في هذا المثال اردنا ان ناتي بانواع الكاتوجوري مع من يرتبط معهم من العملاء

والطريقة نقوم بكتابة الجدول المراد جلب كل اعمدتها اولا ثم اسم الدالة ادناه

ومن ثم نكتب اسم جدول اليمين وبعدها نكتب عمود البرايماري كيي وبعده نكتب الفورين كي

ومن ثم النتيجة تكون بارجاع لكل كاتوجوري معين قائمة كاملة باسم كل العملاء المرتبطين به

وعند طباعة العملاء نعمل زي المثال الي تحت

```
var result = cats.GroupJoin(custs, cat => cat.id, cust => cust.categoryid,
  (cats, custs) => new
  {
    MyCustomers = custs,
    catogeryName = cats.name
  });

  foreach (var res in result)
  {
    Console.WriteLine(res.catogeryName);
```

```
    if (res.MyCustomers != null)
    {
        foreach (var cat in res.MyCustomers)
        {
            Console.WriteLine("------------>" + cat.name);
        }
    }
}
```

وهذه الطريقة لجلب البيانات من الجدول اليمين مع المشترك من جدول اليسار

```
var result = custs.GroupJoin(cats, cust => cust.categoryid, cat => cat.id,
    (custs, cats) => new
    {
        MyCatogeries = cats,
        CustomerName = custs.name
    });

    foreach (var res in result)
    {
        Console.WriteLine(res.CustomerName);
        if (res.MyCatogeries != null)
        {
            foreach (var cat in res.MyCatogeries)
            {
                Console.WriteLine("------------>" + cat.name);
            }
        }
    }
```

وهذه بطريقة الكويري نفس الميثود السابقة

```
var result = from cat in cats
             join cust in custs
             on cat.id equals cust.categoryid
             into Customers
             select new
             {
                 MyCustomers = Customers,
                 catName = cat.name
             };
```

هذه الطريقة لعمل ترتيب للعناصر على حسب عمود معين
في هذا المثال يتم ترتيب العملاء على حسب الكاتوجوري

```
var result = custs.GroupBy(c => c.categoryid);

   foreach (var item in result)
   {
      Console.WriteLine($"number :{item.Key}");

      foreach (var g in item)
      {
         Console.WriteLine("---------->" + g.name);
      }
   }
```

هذا المثال نفس طريقة عمل الجروب باي الماضيه لاكن بدلا الجروب باي سوينا تولوكاب
والفرق بين الجروب باي والتولوكاب هو انه الجروب باي تنفيذه يكون مؤجل لا يتم ملئ قائمة الريزولت
الا عند الفور لوب على الريزولت وبالتالي اي تعديل على مصدر الريزولت اي الجدول المراد عمل عليه الترتيب فهو ستحدث على الريزولت طالما لم يواجهه فور لوب
بينما اللوكاب عكس ذلك الريزولت يتم اسناد لها القيمه مباشره بعد عمل دالة اللوكاب
واي تغيير على الجدول المصدر لا يتاثر به قائمة الريزولت

```
var result = custs.ToLookup(c => c.categoryid);

   foreach (var item in result)
   {
      Console.WriteLine($"number :{item.Key}");

      foreach (var g in item)
      {
         Console.WriteLine("---------->" + g.name);
      }
   }
```

وهذه باستخدام الكويري لكلا الحالتين السابقتين

```
var result = from c in custs
         group c by c.categoryid;

   var result2 = (from c in custs select c).ToLookup(c => c.categoryid);
```

هذه الميثود لعمل قائمة متسلسلة تقوم انت بتحديد بدايتها وعدد عناصرها

```
   var r = Enumerable.Range(1, 100);
```

هذه الدالة تقوم بعمل قائمة فارغة الداتا تايب الي انت تريده

```
   var e = Enumerable.Empty<Customer>();
```

هذه الدالة تقوم بالتكرار اول حاجه تمرر لها العنصر المراد تكراره ومن ثم تمرر عدد مرات التكرار

```
var cust = GetData.GetCustomers().FirstOrDefault();
var rep = Enumerable.Repeat(cust, 4);
```

هذه الدالة نستخدمها لتفكيك القائمة النصية مثلا ولها استخدامات غيرها

```
List<string> names = new List<string>()
{
  "Ahmed mohamady","ali salim","ala khalid"
};

  var res = names.SelectMany(element => element.Split(' '));

  foreach (var name in res)
  {
    Console.WriteLine(name);
  }
```

هذه نفس السابقة بس بطريقة الكويري

```
var res = from c in names
      from ns in c.Split(' ')
      select ns;
```

هذه الدالة تكون بجمع العناصر المشتركة بين قائمتين

```
var l1 = Enumerable.Range(0, 10);
  var l2 = Enumerable.Range(5, 15);
  var l3 = l1.Union(l2);
```

هذه الدالة تقوم بجمع جميع العناصر من العناصر المشتركة وغير المشتركة القائمتين وتضعهم في قائمة جديدة

```
var l1 = Enumerable.Range(0, 10);
  var l2 = Enumerable.Range(5, 15);
  var l3 = l1.Concat(l2);
```

هذه الدالة تقوم باستبعاد جميع الارقام المتكررة من القائمة

```
  var l3 = l1.Distinct();
```

هذه الدالة تقوم بارجاع لي العناصر الموجودة فقط في الليست ون وليست موجوده في الليست تو

```
var l1 = Enumerable.Range(0, 10);
  var l2 = Enumerable.Range(5, 15);
  var l3 = l1.Except(l2);
```

هذه الدالة تقوم بارجاع العناصر المشتركة بين الليست الاولى والليست الثانية

```
  var l1 = Enumerable.Range(0, 10);
```

```
   var l2 = Enumerable.Range(5, 15);
   var l3 = l1.Intersect(l2);
```

هذه الدالة تقوم بارجاع لك اما ترو او فولس وطريقة عملها انها ترجع الناتج
بناء على ان جميع العناصر يحققون الشرط الذي مررته انت بين القوسين

```
var isAll = GetData.GetCustomers().All(x => x.age > 30);
```

هذه الدالة نفس السابقة بس مش شرط تكون كل العناصر محققه الشرط اهم شي
لو حتى عنصر واحد محقق الشرط بيكون الناتج ترو

```
var isAll = GetData.GetCustomers().Any(x => x.age > 30);
```

هذه الدالة تحقق من انه هل القائمة تحتوي على الي تم تمريرها لها بين القوسين
نلا حظ اننا قمنا بتمرير اوبجيكت من قائمة العملاء بين قوسين هذه الدالة وكانت النتيجة
فولس مع انه الاوبجيكت موجود فعليا في القائمة والسبب لانه يعتبره انيستانس جديد

```
var cust = new Customer { id = 101, name = "ahmed mohmed", age = 30, isActive = true,
joinDate = new DateTime(2022, 10, 15), categoryid = 1, spendAverage = 1500.9m, telephone =
123456789 };
   var contai = GetData.GetCustomers().Contains(cust);
   Console.WriteLine(contai.ToString());
```

ولحل هذه المشكلة نقوم بعمل كلاس جديد يورث من كلاس اي ايكواليتي كومبارير
وهذه الكلاس سيطلب منك عمل ايمبليمنتايشن لثنتين ميثود
الاولى نقوم فيها بكتابة الاتريبيوت المراد مقارنتها مع الاوبجيكت والتحقق منها
احنا في مثالنا اكتفينا بالمقارنة عن طريق الاي دي فقط
لاكن بامكانك ان تجعل المقارنة على مستوى جميع العناصر باضافة شروطها
والميثود الثانية تطلب منا ارجاع الهاش كود لقيمه معينه والافضل انه نرجع قيمة الاي دي
لانه فريد وغير متغير
حلو الان قمنا بعمل الكلاس الذي سيمكننا من عمل المقارنة نجي للميثود
كونتاينس التي سنتحقق بها اذا معنا تحقق لداتا تايب عادي مش اوبجيكت
زي الانتيجر والاسترينج وغيرها نمرر للدالة باراميتر واحد وهي بتحقق منه
اما في حالة تحقق من الاوبجيكت لازم نمرر لها الاوبجيكت والباراميتر الثاني اوبجيكت من الكلاس الي عملناه
كما هو موضح في الكود ادناه

```
var cust = new Customer { id = 101, name = "ahmed mohmed", age = 30, isActive = true,
joinDate = new DateTime(2022, 10, 15), categoryid = 1, spendAverage = 1500.9m, telephone =
123456789 };
   var contai = GetData.GetCustomers().Contains(cust, new CustCompare());
   Console.WriteLine(contai.ToString());

public class CustCompare : IEqualityComparer<Customer>
{
   public bool Equals(Customer? x, Customer? y)
   {
      return x.id == y.id;
```

```
    }

    public int GetHashCode([DisallowNull] Customer obj)
    {
        return obj.id.GetHashCode();
    }
}
```

هذه الدالة تقوم بعمل ضغط او جمع بين قائمتين وتجعلهم كعنصر واحد بالنسبه لعناصر القائمتين

```
var nums = Enumerable.Range(0, GetData.GetCustomers().Count);

var result = GetData.GetCustomers().Zip(nums, (c, i) => new
{
    i,
    c.name
});

foreach (var c in result)
{
    Console.WriteLine(c.i.ToString() + " : " + c.name);
}
```

الناتج بيكوون

```
0 : ahmed mohmed
1 : salah ahmed
2 : mahoude samy
3 : mostafa kamel
4 : seed nabil
5 : fareed sif
6 : abdelrahman
7 : sayed kabaka
8 : nora hazem
9 : samy tony
10 : ahmed galal
```

هذه الدالة تتحقق من التسلل لعناصر القائمة
ترجع ترو لو كانت عناصر القائمتين متشابهتين وكان الداتا تايب عادي للقائمتين
زي الانتيجر والاسترينج وغيره اما اذا كان نوع القائمة اوبجيكت من كلاس معين
فينبغي علينا عمل الكلاس اي كوانتي كومبارير الذي تم عمله في الامثلة السابقة
وتمريره في دالة السيكوينسيس

```
List<int> l1 = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
List<int> l2 = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

```
var result = l1.SequenceEqual(l2);

Console.WriteLine(result);
```

هذه الدالة تقوم بعمل دوال التجميع في الاس كيو ال سيرفير من جمع ومتوسط وغيره
في ابسط سيناريو لها نقوم بتمرير لها اثنين متغيرات الاول يمثل القيمة البدائية
والمتغير الثاني يمثل القيمة الثانية في القائمة
وبعدين نسوي اشارة اللمدا ونقول ايش الي نبغاه بالضبط هل نبغى المجموع
او حاصل ضرب العناصر او اي لوجيك انت تريده
لاكن لازم تعرف انه هذه الدالة ترجع بس قيمه وحده
مثلا في هذا المثال نبغى حاصل ضرب جميع عناصر القائمة

```
var numbers = new List<int> { 1, 2, 3, 4, 5 };
int result = numbers.Aggregate((acc, next) => acc * next);
```

في السيناريو الثاني نحدد له مثلا اول قيمه لازم يبدأ بها
ونكتبها قبل المتغيرات الي سويناها قبل
في مثالنا هذا اول حاجه بيجمع اول عنصر مع العشره وبعدين بيجمعها مع باقي العناصر

```
var numbers = new List<int> { 1, 2, 3 };
int result = numbers.Aggregate(10, (acc, next) => acc + next);
```

الان نجي لكتابة هذه الدالة التجميعيه بسيناريوهات متقدمه
وهو اننا بعد كتابة اسم الداله داخل علامتي الاكبر والاصغر نمرر ثلاثه متغيرات
المتغير الاول يمثل ايش نوع عناصر القائمة الي انت بتسوي عليهم العملية
والثاني يمثل نوع التجميع ايش بيكون هل هو انتيجر او دبل او سترينج وغيرها
والثالث يمثل النتيجة النهائية من هذه العملية ايش نوعها
في المثال ادناه احنا حددنا نوع عناصر القائمة وهم اوبجيكتات من كلاس العملاء
وقلنا النتيجه نوع التجميع بيكون في متغير من نوع سترينج
وكذلك النتيجه النهائية هي بتكون سترينج
وكذلك قمنا باخباره بان السترينج سيبدا بهذه الجمله التي تم كتابتها كاول باراميتر
وهو م تم شرحهه سابقا
اما ما بعد اشارة اللمدا هو لوجيك طبيعي فكرته بأنه يقوم بحساب خصم كل عميل
عن طريق ضرب السبيند افيراج في الرقم الموضح ادناه
ومن ثم جمع اسم العميل مع النتيجه السابقة داخل المتغير النصي اس
واخيرا خارج علامة اللمدا نقوم بتحديد النتيجه من هذه العملية الي هي نوعها سترينج
كما هو موضح ادناه

```
var data = GetData.GetCustomers().Aggregate<Customer, string, string>(
   "the deserve discounts : ",
   (s, c) =>
   {
```

```
        var discount = c.spendAverage * 0.5m;
        s += $" {c.name}: {discount},";
        return s;
    },
    o => o.Replace(":", "->").TrimEnd(',', ' ')
);
```

هذه الدالة تقوم بعمل تخطي لبعض الاسطر او العناصر في القائمة
عند كتابتها نمرر لهاباراميتر وهو يمثل عدد العناصر التي سيتخطاهم من اول القائمه
وفي مثالنا احنا سيتخطى اول عنصرين في القائمة

```
var list = GetData.GetCustomers().Skip(2);

foreach (var cust in list)
{
    Console.WriteLine(cust.name);
}
```

بامكاننا ان نجعل الدالة السابقة تعمل بناء على شرط
بمعنى العناصر الي تحقق الشرط اعمل لها تخطي

```
var list = GetData.GetCustomers().SkipWhile(x => x.age > 25);

foreach (var cust in list)
{
    Console.WriteLine(cust.name);
}
```

وكذلك نعمل تخطي لعناصر من النهاية تماما مثل م تم عمله في المثال ما قبل السابق

```
var list = GetData.GetCustomers().SkipLast(2);

foreach (var cust in list)
{
    Console.WriteLine(cust.name);
}
```

هذه الدالة نفس السابقة بالضبط مع جميع حالاتها لاكن بدل ما انت تقول
اعمل تخطي لا انت بتقولة خذ فقط

```
var list = GetData.GetCustomers().Take(2);
var list = GetData.GetCustomers().TakeLast(2);
var list = GetData.GetCustomers().TakeWhile(x => x.age > 25);
```

```
foreach (var cust in list)
{
   Console.WriteLine(cust.name);
}
```

هذا المثال بس يشرح استخدام الليت بحيث تسند الكويري لمتغير او حاويه
وتعمل عليها شروطك الي تريدها :-(
طريقة عمل المثال واضحه م تحتاج شرح

```
List<string> names = new List<string>
{
   "@hmed","mo&ed","3alah"
};

var result = from n in names
          let newVal = Regex.Replace(n, "[@&3]", "_")
          where newVal.Contains("m")
          orderby newVal.Length descending
          select newVal;
```

تم الانتهاء من الكووورس بحمد الله وتووووفيقة