List<int> input = new List<int>() { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 };

foreach (int item in input)  
{  
 // 使用线程池线程（后台线程）执行任务  
 Task.Run(() => {  
 Console.WriteLine($"当前数据：{ item } " +  
 $"当前任务Id：{ Task.CurrentId }" +  
 $"当前线程Id：{ Thread.CurrentThread.ManagedThreadId }");  
 });  
}

// 前台线程  
new Task(()=> { Console.WriteLine("这是一个前台线程"); }, TaskCreationOptions.LongRunning).Start();

// 连续任务  
Task continueTask = new Task(() => { Console.WriteLine("任务1"); });  
continueTask.ContinueWith(task => { Console.WriteLine("任务2"); });  
continueTask.Start();

// 取消任务  
CancellationTokenSource cancellationTokenSource = new CancellationTokenSource();  
cancellationTokenSource.Cancel();  
foreach (int item in input)  
{  
 // 当在其他地方调用cts.Cancel();时，未执行的任务不会执行  
 Task.Run(() =>   
 {  
 Console.WriteLine($"当前任务不会执行");  
 }, cancellationTokenSource.Token);  
}

Console.ReadKey();

Task.Run默认使用线程池线程执行任务，线程池线程属于后台线程，当主线程结束后其后台线程也会结束

前台线程主线程结束后后台线程不会结束