```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
namespace FileSaveApp
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter text to save: ");
            string userText = Console.ReadLine();
            Console.Write("Enter file path to save to (example: C:\\Users\\User\\Desktop\\myfile.txt): ");
            string path = Console.ReadLine();
            if (!string.IsNullOrWhiteSpace(userText) && !string.IsNullOrWhiteSpace(path))
            {
                try
                {
                    File.WriteAllText(path, userText);
                    Console.WriteLine("Text saved successfully!");
                }
                catch (Exception ex)
                {
                    Console.WriteLine("Error while saving file:");
                    Console.WriteLine(ex.Message);
                }
            }
            else
            {
                Console.WriteLine("Text or path is empty!");
            }
        }
    }
}
class Program
{
    static void Main()
    {
        Console.Write("Sisesta püramiidi kõrgus: ");
        int rows = int.Parse(Console.ReadLine());
        for (int i = 1; i <= rows; i++)
        {
            // Trüki tühikud ette
            for (int j = i; j < rows; j++)
            {
                Console.Write(" ");
            }
            // Trüki numbrid 1 kuni i
            for (int k = 1; k <= i; k++)
            {
                Console.Write(k + " ");
            }
            Console.WriteLine();
        }
    }
}
namespace LinqExample
{
```

```
    public class Product
    {
        public string Name { get; set; }
        public int Price { get; set; }
    }
    class Program
    {
        static void Main(string[] args)
        {
            List<Product> products = new List<Product>
            {
                new Product { Name = "Phone", Price = 300 },
                new Product { Name = "Laptop", Price = 800 },
                new Product { Name = "Mouse", Price = 20 },
                new Product { Name = "Keyboard", Price = 50 },
                new Product { Name = "Monitor", Price = 200 },
                new Product { Name = "USB Cable", Price = 10 }
            };
            var expensive = products.Where(p => p.Price > 100);
            Console.WriteLine("Products that cost more than 100:");
            foreach (var item in expensive)
            {
                Console.WriteLine($"{item.Name} - ${item.Price}");
            }
            var takeWhileCheap = products.TakeWhile(p => p.Price < 300);
            Console.WriteLine("\nProducts until the first expensive one:");
            foreach (var item in takeWhileCheap)
            {
                Console.WriteLine($"{item.Name} - ${item.Price}");
            }
        }
    }
}
```