|  |
| --- |
| **Анализ на задачата**  Виждаме че в най-широката си част ромба има 2\***N** елемент  На първия ред има два елемента и празни места от ляво и дясно (местата от двете странни са равен брой), отук следва че **N**-1 празни места има на 1ви ред, 1 наклонено няляво и 1 нядясно.  На веки следващ ред до **N-ти** наклонините на едната и другата страна се увеличават с 1. А празнияте места от няво надясно намалят с 1.  Така се изчисляват следните три компонента до **N-ти** ред  string spaces = new string(' ', N - i);  string onLeft = new string('/', i);  string onRight = new string('\\', i);  И се изведат на екрана в последователност spaces,onLeft, onRight с нов ред накрая.  От **N-ти** до 2\*N ред нещата се обърщат. Но ние пак може да го разгледаме като поредица от 1 до N  А за втората част се изчисляват по следния начин  string spaces = new string(' ', N - i);  string onLeft = new string('/', i);  string onRight = new string('\\', i);  И се извеждат в последователнос spaces onRight onLeft с нов ред накрая. |
| **Решение (сорс код)** |
| using System;  namespace Romb  {  class Romb  {  static void Main(string[] args)  {  int N = int.Parse(Console.ReadLine());    for (int i = 1; i <= N; i++)  {  string spaces = new string(' ', N - i);  string onLeft = new string('/', i);  string onRight = new string('\\', i);  Console.Write(spaces);  Console.Write(onLeft);  Console.WriteLine(onRight);  }  for (int i = N; i > 0; i--)  {  string spaces = new string(' ', N - i);  string onLeft = new string('/', i);  string onRight = new string('\\', i);  Console.Write(spaces);  Console.Write(onRight);  Console.WriteLine(onLeft);    }  }  }  } |