A group of n people played a game in which the loser must double the money of the rest of the players. The game has been played n times, and each player has lost exactly once. Surprisingly, each player ended up with the same amount of money of m dollars.

Considering all that, find the amount of money each player had at the very beginning, and return it as an array of n elements sorted in descending order.

**Example**

For n = 3 and m = 16, the output should be  
doubleTheMoneyGame(n, m) = [ 26.0, 14.0, 8.0 ].

Let's say that player A started with $26, player B had $14, and player C had $8.

After the first game, player A lost, and had to pay double the amount of players' B and C money. So the amount of money the players had at the end of the game was [ 4.0, 28.0, 16.0 ].

After the second game, player B lost, and the "money array" became [ 8.0, 8.0, 32.0 ].

After the third game, player C lost, the money became [ 16.0, 16.0, 16.0 ].

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] integer n**

The number of players, aka the number of played games.

*Constraints:*  
2 ≤ n ≤ 7.

* **[input] integer m**

The same amount of money each player has after n game played.

*Constraints:*  
1 ≤ m ≤ 10000.

* **[output] array.float**

The amount of money each player had in the beginning of the game sorted in descending order. It is guaranteed that the values in the output won't have more than 5 digits after the decimal point.

<https://codefights.com/challenge/4Mpg5NjZrFJtqvhGC/main>

static double[] doubleTheMoneyGame(int n, int m)

{

List<double> lista = new List<double>();

for (int i = 0; i < n; i++)

{

lista.Add(m);

}

for (int indice = 0; indice < n; indice++)

{

double resto = 0;

for (int i = 0; i < n; i++)

{

if (i != indice)

{

resto += lista[i] - (lista[i] / 2);

lista[i] /= 2;

}

}

lista[indice] += resto;

}

//for (int i = 0; i < n; i++)

//{

// Console.Write(lista[i] + " ");

//}

lista.Sort();

lista.Reverse();

return lista.ToArray();

}