Given an array of 2k integers (for some integer k), perform the following operations until the array contains only one element:

* On the 1st, 3rd, 5th, etc. iterations (1-based) replace each pair of consecutive elements with their sum;
* On the 2nd, 4th, 6th, etc. iterations replace each pair of consecutive elements with their product.

After the algorithm has finished, there will be a single element left in the array. Return that element.

**Example**

For inputArray = [1, 2, 3, 4, 5, 6, 7, 8], the output should be  
arrayConversion(inputArray) = 186.

We have [1, 2, 3, 4, 5, 6, 7, 8] -> [3, 7, 11, 15] -> [21, 165] -> [186], so the answer is 186.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] array.integer inputArray**

*Constraints:*  
1 ≤ inputArray.length ≤ 20,  
-9 ≤ inputArray[i] ≤ 99.

* **[output] integer**

<https://codefights.com/arcade/code-arcade/spring-of-integration/dwA8RtDF86WucuoaQ>

static int arrayConversion(int[] inputArray)

{

if (inputArray.Length == 1)

{

return inputArray[0];

}

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

int iter = 1;

while (inputArray.Length > 1)

{

lista = new List<int>();

if (iter % 2 != 0)

{

for (int i = 0; i + 1 < inputArray.Length; i += 2)

{

lista.Add(inputArray[i] + inputArray[i + 1]);

}

//if (inputArray.Length % 2 != 0)

//{

// lista.Add(inputArray[inputArray.Length - 1]);

//}

}

else

{

for (int i = 0; i + 1 < inputArray.Length; i += 2)

{

lista.Add(inputArray[i] \* inputArray[i + 1]);

}

//if (inputArray.Length % 2 != 0)

//{

// lista.Add(inputArray[inputArray.Length - 1]);

//}

}

if (inputArray.Length % 2 != 0)

{

lista.Add(inputArray[inputArray.Length - 1]);

}

foreach (int elem in lista)

{

Console.Write(elem + " ");

}

Console.WriteLine();

inputArray = lista.ToArray();

//Console.WriteLine();

iter++;

}

return lista[0];

}