**7 kyu**

**Simple Fun #50: Array Conversion**

1292% of 5410 of144[myjinxin2015](https://www.codewars.com/users/myjinxin2015)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/simple-fun-number-50-array-conversion/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/simple-fun-number-50-array-conversion/solutions/csharp)

[Discourse (6)](https://www.codewars.com/kata/simple-fun-number-50-array-conversion/discuss/csharp)

* Add to Collection
* |
* Share this kata:

**Task**

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 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**

* [input] integer array arr

Constraints: 21 ≤ arr.length ≤ 25, -9 ≤ arr[i] ≤ 99.

* [output] an integer

<https://www.codewars.com/kata/simple-fun-number-50-array-conversion/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public class Kata

{

public int ArrayConversion(int[] arr)

{

//coding and coding..

int b = 0;

//coding and coding..

while (arr.Length > 1)

{

var a = new int[arr.Length / 2];

var cnt = 0;

for (int i = 0; i < arr.Length; i++)

{

a[cnt] = b % 2 == 0 ? arr[i] + arr[i + 1] : arr[i] \* arr[i + 1];

cnt++;

i++;

}

b++;

arr = a;

}

return arr[0];

}

}

static int[] SumarArray(int[] arr)

{

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

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

{

ans.Add(arr[i] + arr[i + 1]);

}

return ans.ToArray();

}

static int[] MultiplicarArray(int[] arr)

{

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

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

{

ans.Add(arr[i] \* arr[i + 1]);

}

return ans.ToArray();

}

public static int ArrayConversion(int[] arr)

{

//coding and coding..

int[] res = arr.ToArray();

//int len\_res = 0;

//int len\_anterior = arr.Length;

while(true)

{

//len\_anterior = res.Length;

res = SumarArray(res);

if (res.Length == 1) return res[0];

res = MultiplicarArray(res);

if (res.Length == 1) return res[0];

//len\_res = res.Length;

}

//foreach(int item in res)

//{

// Console.Write(item + " ");

//}

//Console.WriteLine();

return res[0];

}

static void Main(string[] args)

{

int[] arr = { 1, 2 };

int res = ArrayConversion(arr);

Console.WriteLine(res);

Console.ReadLine();

}

}

}