* **Task**

You're given an arbitrary 32-bit integer n. Swap each pair of adjacent bits in its binary representation and return the result as a decimal number.

* **Example**

For n = 13, the output should be 14

1310 = 11012 ~> 11102 = 1410

For n = 74, the output should be 133

7410 = 010010102 ~> 100001012 = 13310

* + Note

the preceding zero written in front of the initial number: since both numbers are 32-bit integers, they have 32 bits in their binary representation. The preceding zeros in other cases don't matter, so they are omitted. Here, however, it does make a difference.

* **Input/Output**
  + [input] integer n

Constraints: 0 ≤ n < 230.

* + [output] an integer

<http://www.codewars.com/kata/simple-fun-number-11-swap-adjacent-bits/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

//----------MI SOLUCION--------------

public static int SwapAdjacentBits(int n)

{

//coding and coding..

string bin = Convert.ToString(n, 2);

while (bin.Length < 32)

{

bin = '0' + bin;

}

char[] b = bin.ToCharArray();

//Console.WriteLine(b);

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

{

char temp = b[i];

b[i] = b[i + 1];

b[i + 1] = temp;

}

return Convert.ToInt32(new string(b), 2);

}

static void Main(string[] args)

{

//SwapAdjacentBits(5);

//Console.WriteLine( Convert.ToInt32("101", 2));

Console.WriteLine(SwapAdjacentBits(83748));

//1101

//1110

Console.ReadLine();

}

}

}

---------solucion por [Kaiyou](http://www.codewars.com/users/Kaiyou) --------------

public class Kata

{

public int SwapAdjacentBits(int n)

{

string binary = Convert.ToString(n, 2);

string swapped = "";

while (binary.Length < 32)

binary = "0" + binary;

for (int i = 1; i<=32; i+=2)

swapped+= "" + binary[i] + binary[i-1];

return Convert.ToInt32(swapped, 2);

}

}