Given an array of non-negative integers arr, your task is to find and return the position of the first number in the array that has the longest number of consecutive set bits in its binary representation.

For instance, the longest number of consecutive bits in 11 is 2, since 1110 = 10112.

**Example**

For arr = [1, 2, 3, 4, 5, 6], the output should be  
longestConsecutive(arr) = 2.

With all its values converted to binary, arr turns into ['1', '10', '11', '100', '101', '110']. As you can see, 3 and 6 have the longest streak of consecutively set bits (2). Since 3 comes before 6 in the array, and its position is 2, the output should be 2.

**Input/Output**

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

An array of non-negative integers.

*Guaranteed constraints:*  
1 ≤ arr.length ≤ 105,  
0 ≤ arr[i] < 231.

* **[output] integer**

The position of the first number in the array that has the longest number of consecutive set bits in its binary representation.

<https://codefights.com/challenge/J9z6mzQQziyZuwPQJ?utm_source=emailNotification&utm_medium=email&utm_campaign=featuredChallenge>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class Program

{

static int longestConsecutive(int[] arr)

{

int max\_pos = 0;

int max\_unos = 0;

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

{

string bin = Convert.ToString(arr[i], 2);

//Console.WriteLine(bin + " ");

int j = 0;

while (j < bin.Length)

{

int unos = 0;

while (j < bin.Length && bin[j] == '1')

{

unos++;

j++;

}

j++;

//max\_unos = Math.Max(max\_unos, unos);

if (unos > max\_unos)

{

max\_unos = unos;

max\_pos = i;

}

}

}

return max\_pos;

}

static void Main(string[] args)

{

int[] arr = { 1, 2, 3, 4, 5,6 };

Console.WriteLine( longestConsecutive(arr));

Console.ReadLine();

}

}

}