Given a non-negative integer k, convert it to binary, swap the bits at positions p1 and p2counting from the right, and return the obtained number in base 10.

**Example**

For k = 565, p1 = 9 and p2 = 3, the output should be  
swapPairBits(k, p1, p2) = 817.

k2 = 1**0**00110**1**01 (Here, p1 is colored **red** and p2 is colored **blue**.) With p1 and p2swapped, k becomes 1**1**00110**0**01, which is 81710. Thus, the final answer is 817.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] integer64 k**

*Guaranteed constraints*  
0 ≤ k ≤ 250.

* **[input] integer p1**

*Guaranteed constraints:*  
1 ≤ p1 ≤ 51.

* **[input] integer p2**

*Guaranteed constraints:*  
1 ≤ p2 ≤ 51.

* **[output] integer64**

The number obtained by swapping the bits at p1 and p2.

<https://codefights.com/challenge/EYpoYCY6FNTKeimZx?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 long swapPairBits(long k, int p1, int p2)

{

string s = Convert.ToString(k,2);

while (s.Length < Math.Max(p1, p2))

{

s = '0' + s;

}

char[] bin = s.ToCharArray();

// Console.WriteLine(bin.Length);

// Console.WriteLine(new string(bin));

int a = bin.Length - p1;

int b = bin.Length - p2;

// Console.WriteLine(a);

// Console.WriteLine(b);

char temp = bin[a];

bin[a] = bin[b];

bin[b] = temp;

return Convert.ToInt64( new string(bin),2 );

}

static void Main(string[] args)

{

// int k = 565, p1 = 9 , p2 = 3;

long k= 512;

int p1= 33;

int p2 = 10;

Console.WriteLine(swapPairBits(k, p1, p2));

Console.ReadLine();

}

}

}