Author

[ravst](https://codefights.com/profile/ravst)

https://codefights.com/img/coins_new.png2000

Return a list of all numbers that have no more than n bits, such that exactly k of them are set to 1.

**Example**

For n = 4 and k = 1, the output should be  
kBitsDigits(n, k) = [1,2,4,8].

110 = 12, which obviously has 1 < 4 number of bits, and the only bit is 1.  
210 = 102, which has 2 < 4 number of bits, with the first one equal to 1.  
410 = 1002, which has 3 < 4 number of bits, with the first one equal to 1.  
810 = 102, which has 3 < 4 number of bits, with the first one equal to 1.

* **[input] integer n**

The maximum number of bits the number can have, 1 ≤ n ≤ 20.

* **[input] integer k**

The number of bits equal to 1, 0 ≤ k ≤ 20.

* **[output] array.integer**

The numbers with at most n bits k of which are set to 1 sorted in ascending order.

<https://codefights.com/challenge/uwTrqGQj8iE3xeibW>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int[] kBitsDigits(int n, int k)

{

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

if (k == 0)

{

answer.Add(0);

return answer.ToArray();

}

int cont = 0;

for (int i = 1; cont < n; i++)

{

int c = i;

cont = 0;

while (c > 0)

{

if (c % 2 == 1) {

cont++;

}

c /= 2;

}

if (cont == k)

{

answer.Add(i);

}

}

return answer.ToArray();

}

static void Main(string[] args)

{

foreach (int elem in kBitsDigits(4, 1))

{

Console.Write(elem + " ");

}

Console.ReadLine();

}

}

}