Given an integer n, you can remove either its first or last digit in one step. After applying this operation several times, you'll get a number x with a length of k. It's possible that the number will contain leading zeros. What is the minimal and the maximal possible value of x that you can obtain?

**Example**

* For n = 15243 and k = 2, the output should be  
  removeDigits(n, k) = [15, 52].
  + To obtain the minimal number with a length of k, we can use the following sequence of operations: 15243 -> 1524 -> 152 -> 15;
  + To obtain the maximal number with a length of k, we can use the following sequence of operations: 15243 -> 1524 -> 152 -> 52.
* For n = 123 and k = 1, the output should be  
  removeDigits(n, k) = [1, 3].

**Input/Output**

* **[time limit] 6000ms (cs)**
* **[input] integer64 n**

The initial number.

*Guaranteed constraints:*  
10 ≤ n ≤ 1015.

* **[input] integer k**

The desired length of the output number.

*Guaranteed constraints:*  
1 < 10k ≤ n.

* **[output] array.integer64**

An array in which the first element is equal to the minimal possible number with length k and the second element is equal to the maximal possible number of length k.

<https://codefights.com/challenge/mYSs9Exk9t7YCmfua?utm_source=emailNotification&utm_medium=email&utm_campaign=featuredChallenge>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static long[] removeDigits(long n, int k)

{

string ns = n.ToString();

long min = long.MaxValue;

long max = long.MinValue;

for (int i = 0; i < ns.Length - k+1; i++)

{

//Console.WriteLine(ns.Substring(i, k));

long num = long.Parse(ns.Substring(i, k));

min = Math.Min(min, num);

max = Math.Max(max, num);

}

return new long[] { min, max };

}

static void Main(string[] args)

{

//long n = 1234567;

//int k = 3;

long n = 15243 ;

int k = 2;

foreach (long l in removeDigits(n, k))

{

Console.Write(l + " ");

}

Console.ReadLine();

}

}

}