Given a positive integer remove K digits from it to make the number as small as possible. If after removing digits the number has leading zeros then these zeros are also removed (for example, number 23 can be obtained from 1023 by removing only digit '1').

* **[time limit] 3000ms (cs)**
* **[input] integer N**

positive integer, N < 10^9

* **[input] integer K**

positive integer, K is less than the number of digits in N

* **[output] integer**

smallest obtainable number

<https://codefights.com/challenge/HSr75XNZSgoxyKxDX/main>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static List<List<int>> Powerset(int[] nums)

{

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

ps.Add(new List<int>()); // add the empty set

// for every item in the original list

foreach (int item in nums)

{

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

foreach (List<int> subset in ps)

{

// copy all of the current powerset's subsets

newPs.Add(subset);

// plus the subsets appended with the current item

List<int> newSubset = new List<int>(subset);

newSubset.Add(item);

newPs.Add(newSubset);

}

// powerset is now powerset of list.subList(0, list.indexOf(item)+1)

ps = newPs;

}

return ps;

}

static int digitsRemoving(int N, int K)

{

char[] ch = N.ToString().ToCharArray();

int[] digitos = Array.ConvertAll(ch, e => int.Parse(e.ToString()));

//foreach (int elem in digitos)

//{

// Console.Write(elem + " ");

//}

List<List<int> > ps = Powerset(digitos);

int min = int.MaxValue;

foreach (List<int> lista in ps)

{

if (lista.Count == N.ToString().Length - K )

{

var result = string.Join("", lista.Select(x => x.ToString()).ToArray());

if (result.Length > 0)

{

int ent = int.Parse(result);

min = Math.Min(min, ent);

//Console.WriteLine(ent);

}

}

}

return min;

}

static void Main(string[] args)

{

Console.WriteLine( digitsRemoving(45678, 2) );

//int[] arr = { 1, 2, 3 };

//var result = string.Join("", arr.Select(x => x.ToString()).ToArray());

//Console.WriteLine(int.Parse( result)+1);

Console.ReadLine();

}

}

}

------------------SOLUCION POR Justin---------------------------

static long digitsRemoving(int N, int K)

{

StringBuilder s = new StringBuilder(N + "");

for (int i; K-- > 0; )

{

Console.WriteLine(s);

for (i = 0; ++i < s.Length && s[i - 1] <= s[i]; ) ;

s.Remove(i - 1, 1);

}

return long.Parse(s + "");

}