

题目描述：

给一个正整数 NUM1，计算出新正整数 NUM2，NUM2 为 NUM1 中移除 N 位数字后的结果，需要使得 NUM2 的值最小。

输入描述：

1.输入的第一行为一个字符串，字符串由 0-9 字符组成，记录正整数 NUM1，NUM1 长度小于 32。

2.输入的第二行为需要移除的数字的个数，小于 NUM1 长度。

如：

2615371

4

输出描述：

输出一个数字字符串，记录最小值 NUM2。

如：131

补充说明：

```
示例1
输入：2615371
      4
输出：131
说明：移除2、6、5、7这四个数字，剩下1、3、1按原有顺序排列组成131，为最小值
```

```

1 import java.util.Scanner;
2
3 // 注意类名必须为 Main, 不要有任何 package xxx 信息
4 public class Main {
5     private static String[][] dp = new String[33][33];
6
7     public static void main(String[] args) {
8         Scanner in = new Scanner(System.in);
9         // 注意 hasNext 和 hasNextLine 的区别
10        while (in.hasNextLine()) { // 注意 while 处理多个 case
11            String a = in.nextLine();
12            int b = Integer.parseInt(in.nextLine());
13            int[] map = new int[a.length()];
14            for (int i = 0; i < a.length(); i++) {
15                map[i] = a.charAt(i) - '0';
16            }
17            int max = a.length() - b;
18            String res = func(map, 0, 0, max);
19            int l = 0;
20            while (l < res.length() && res.charAt(l) == '0') l++;
21            if (l >= res.length()) System.out.println(0);
22            else System.out.println(res.substring(l));
23        }
24    }
25
26    public static String func(int[] map, int i, int j, int max) {
27        if (i == max) return "";
28        if (dp[i][j] != null) return dp[i][j];
29        StringBuilder tmp = new StringBuilder();
30        for (int k = 0; k < max - i; k++) tmp.append("9");
31        String res = tmp.toString();
32        for (int k = j; k < map.length; k++) {
33            if (max - i > map.length - k) break;
34            String cur = String.valueOf(map[k]) + func(map, i + 1, k + 1, max);
35            if (res.compareTo(cur) > 0) res = cur;
36        }
37        dp[i][j] = res;
38        return res;
39    }
40 }

```