

题目描述：

对于一个连续正整数组成的序列，可以将其拼接成一个字符串，再将字符串里的部分字符打乱顺序。如序列 `8 9 10 11 12`，拼接成的字符串为 `89101112`，打乱一部分字符后得到 `90811211`。注意打乱后原来的正整数可能被拆开，比如在 `90811211` 中，原来的正整数 `10` 就被拆成了 `0` 和 `1`。

现给定一个按如上规则得到的打乱了字符的字符串，请将其还原成连续正整数序列，并输出序列中最小的数字。

输入描述：

输入一行，为打乱字符的字符串和正整数序列的长度，两者间用空格分隔，字符串长度不超过 `200`，正整数不超过 `1000`，保证输入可以还原成唯一序列。

输出描述：

输出一个数字，为序列中最小的数字。

补充说明：

示例 1

输入：

19801211 5

输出：

8

说明：

还原出的序列为 `8 9 10 11 12`，故输出 `8`

示例 2

输入：

432111111111 4

输出：

111

说明：

还原出的序列为 *111 112 113 114*，故输出 *111*

```
import java.util.Arrays;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
        int n = sc.nextInt();

        char[] arr = str.toCharArray();
        Arrays.sort(arr);
        String s = new String(arr);
        for (int i = 1; i < 10000; i++) {
            if (s.equals(getStr(i, n))) {
                System.out.println(i);
                break;
            }
        }
    }

    public static String getStr(int k, int n) {
        String res = "";
        for (int i = 0; i < n; i++) {
            res += k;
            k++;
        }
        char[] chs = res.toCharArray();
        Arrays.sort(chs);
        return new String(chs);
    }
}
```