

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：

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

```
/**
```

```
 * @作者 Brown
```

```
 * @日期 2023/6/21 16:12
```

```
 */
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        String s = sc.next();
```

```
        int n = sc.nextInt();
```

```
        int[] f = new int[10], g = new int[10];
```

```
        for (int i = 0; i < s.length(); i++) {
```

```
            f[s.charAt(i) - '0']++;
```

```
        }
```

```
        for (int i = 1; i < n; i++) {
```

```
            op(i, 1, g);
```

```
        }
```

```
        for (int i = n; ; i++) {
```

```
            op(i, 1, g);
```

```
            op(i - n, -1, g);
```

```
            if(get(g, f)){
```

```
                System.out.println(i-n+1);
```

```
                break;
```

```
            }
```

```
        }
```

```
    }
```

```
    public static boolean get(int[] g, int[] f) {
```

```
        for (int i = 0; i < f.length; i++) {
```

```
        if (f[i] != g[i])
            return false;
    }
    return true;
}

public static void op(int x, int st, int[] g) {
    while (x > 0) {
        g[x % 10] += st;
        x /= 10;
    }
}

}
```