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

```
#include <bits/stdc++.h>
using namespace std;


string s;
int len;
int num[15] = {0};
int tmp[15] = {0};

bool ok(int k) {
	for (int i = 0; i < 10; i++) tmp[i] = 0;
	for (int i = k; i < k + len; i++) {
		int x = i;
		while (x > 0) {
			tmp[x % 10]++;
			x /= 10;
		}
	}
	for (int i = 0; i < 10; i++) if(num[i] != tmp[i]) return 0;
	return 1;
}
int main() {
	// ios :: sync_with_stdio(0);
	cin >> s >> len;
	for(auto& c : s) num[c-'0']++;
	for(int i = 0; i <= 10000; i++) {
		if(ok(i)) {
			cout << i << endl;
			break;
		}
	}
	return 0;
}
// 64 位输出请用 printf("%lld")
```