

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: 492111111111 4
输出: 111
说明: 还原出的序列为111 112 113 114, 故输出111
```

```
1 #include <iostream>
2 #include<algorithm>
3 #include <string>
4 #include<cmath>
5 using namespace std;
6
7 bool check(int x,int len,string ss){
8     string tar="";
9     for(int i=x;i<=x+len;i++){
10         tar+=to_string(i);
11     }
12     sort(tar.begin(),tar.end());
13     if(tar==ss){
14         return true;
15     }
16     return false;
17 }
18 int main() {
19     string ss;
20     int len;
21     cin>>ss>>len;
22     sort(ss.begin(),ss.end());
23     int cnt = ss.length()/len;
24     int start = pow(10, cnt-1);
25     for(int i=start;i<=start*10;i++){
26         if(check(i,len,ss)){
27             cout<<i<<endl;
28             break;
29         }
30     }
31     return 0;
32 }
33 // 64 位输出请用 printf("%lld")
```