找最小数

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

```
#include <iostream>
#include <string>
using namespace std;
void dfs(string& dec,string str,int curindex)
{
    int len=str.size();
    if(len==0 || curindex==len)
    {
        return;
    }
    if(curindex==0)
    {
        dec+=str;
        return ;
    }
```

```
    int index=0;
    for(int i=0;i<curindex+1 && i<len;++i)
    {
        if(str[i]<str[index])
        {
            index=i;
        }
    }
    dec+=str[index];
    if(index+1 == len)
    {
        return;
    }
    dfs(dec,str.substr(index+1),curindex-index);//2615371 5371 71
    return;
}
int main()
{
    string str;
    cin>>str;
    int num;
    cin>>num;
    string dec="";
    dfs(dec,str,num);
    int size=dec.size();
    int index=0;
    //排除 0 在不合法位置情况
    for(;index<size;++index)
    {
        if(dec[index]!='0')
        {
            break;
        }
    }
    if(index==size)
    {
        cout<<0;
    }else{
        for(;index<size;++index)
        {
            cout<<dec[index];
        }
        cout<<endl;
    }
```

```
    return 0;
}
```