

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

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

```
#include <iostream>
```

```
#include <stack>
```

```
#include <string>
```

```
using namespace std;
```

```

int main() {

    string s,ans="";

    int n,x;

    cin>>s;

    cin>>n;

    stack<int> st;

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

        x=s[i]-'0';

        while(!st.empty()&&n&&st.top()>x){

            st.pop();

            n--;

        }

        if(st.empty()&&x==0)

            continue;

        st.push(x);

    }

    while(n&&!st.empty()){

        n--;

        st.pop();

    }

    if(st.empty())

        ans="0";

    while(!st.empty()){

        x=st.top();

        st.pop();

        ans=(char)(x+'0')+ans;

    }
}

```

```
cout<<ans;  
}  
// 64 位输出请用 printf("%lld")
```