

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

int main() {
    int b;
    string s;
    cin >> s >> b;
    int n = s.size();
    // cout << s << endl;
    // cout << b << endl;

    vector<int> res;

    for (int i = b, last = 0; i < n; i++) {
        // find min num idx in [last, i] t
        int t = -1;
        for (int j = last; j <= i; j++) {
            if (t == -1 || s[j] < s[t]) t = j;
        }
        last = t + 1;
        res.push_back(s[t] - '0');
    }

    // strip leading zero
    reverse(res.begin(), res.end());
    while (res.size() && res.back() == 0) res.pop_back();
    for (int i = res.size() - 1; i >= 0; i--) cout << res[i];
    if (res.empty()) cout << 0;
    cout << endl;

    return 0;
}

```