

题目描述：

单词接龙的规则是：可用于接龙的单词首字母必须要前一个单词的尾字母相同；当存在多个首字母相同的单词时，取长度最长的单词，如果长度也相等，则取字典序最小的单词；已经参与接龙的单词不能重复使用。

现给定一组全部由小写字母组成单词数组，并指定其中的一个单词作为起始单词，进行单词接龙，请输出最长的单词串，单词串是单词拼接而成，中间没有空格。

输入描述：

输入的第一行为一个非负整数，表示起始单词在数组中的索引 K ， $0 \leq K < N$ ；

输入的第二行为一个非负整数，表示单词的个数 N ；

接下来的 N 行，分别表示单词数组中的单词。

输出描述：

输出一个字符串，表示最终拼接的单词串。

补充说明：

单词个数 N 的取值范围为 $[1, 20]$ ；

单个单词的长度的取值范围为 $[1, 30]$ ；

示例 1

输入：

```
0
6
word
dd
da
dc
dword
d
```

输出：

wordddworddda

说明：

先确定起始单词 *word*，再接以 *d* 开头的且长度最长的单词 *dword*，剩余以 *d* 开头且长度最长的有 *dd*、*da*、*dc*，则取字典序最小的 *da*，所以最后输出 *wordddworddda*。

示例 2

输入：

4
6
word
dd
da
dc
dword
d

输出：

dworddda

说明：

先确定起始单词 *dword*，剩余以 *d* 开头且长度最长的有 *dd*、*da*、*dc*，则取字典序最小的 *da*，所以最后输出 *dworddda*。

```
import java.util.*;
```

```
public class Main {  
    static int[] vis;  
    static String[] s;  
    static int m,n;  
    static String ans = "";  
  
    static class node {  
        String p;  
        int index;  
        node(String p, int index) {this.p = p; this.index = index;}  
    }  
    static void dfs(String now){
```

```

        char ch = now.charAt(now.length() - 1);
        List<node> arr = new ArrayList<>();
        for (int i = 0; i < n; i++) {
            if (vis[i] == 0 && s[i].charAt(0) == ch) {
                arr.add(new node(s[i], i));
            }
        }
        arr.sort((x, y) -> {
            if (x.p.length() != y.p.length()) {
                return y.p.length() - x.p.length();
            } else {
                return x.p.compareTo(y.p);
            }
        });
        if (arr.size() > 0) {
            vis[arr.get(0).index] = 1;
            ans += arr.get(0).p;
            dfs(arr.get(0).p);
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        m = sc.nextInt();
        n = sc.nextInt();
        vis = new int[n];
        s = new String[n];
        for (int i = 0; i < n; i++) {
            s[i] = sc.next();
        }
        ans += s[m];
        vis[m] = 1;
        dfs(s[m]);
        System.out.println(ans);
    }
}

```