

## 词链

### 【问题描述】

一个词是由至少 1 个，至多 75 个小写英文字母(a..z)组成。当在一张由一个或多个词组成的表中，每一个词（除第一个外）都能由在其前一个词的词尾添加一个或多个字母而得到，则称此表为一个链。

例如：

*i*

*in*

*int*

*integer*

为一个含 4 个词的词链，而表

*input*

*integer*

不是词链。注意：所有含有一个词的表都是链。

给定一个词按字典顺序由小到大排列的表，找出表中的最长词链。表的大小最大达到 2M。

### 【算法分析】

这道题目，虽然测试数据大得惊人，但是难度却并不大。主要的是要把握住表的特点：词按字典顺序由小到大的排列，因此表中的相邻的两个词语之间的关系只有两种——属于同一词链或不属于同一词链，若它们不属于同一词链，那么后一个单词所在的词链，除这一个单词外的链（这个链可能为空）必定为它在表中的前一个单词的词链的前缀。举个例子：

*i, in, int, integer, inter*

在上面这个例子当中，最后一个单词 *inter* 所组成的词链为  $i \rightarrow in \rightarrow int \rightarrow inter$ ，在这个链中，除 *inter* 的外的链为  $i \rightarrow in \rightarrow int$ 。而它的前一个单词 *integer* 组成的词链为  $i \rightarrow in \rightarrow int \rightarrow integer$ 。其中  $i \rightarrow in \rightarrow int$  就是  $i \rightarrow in \rightarrow int \rightarrow integer$  中的前缀。

因此，可以利用堆栈这种数据结构来解决这一问题。以下是算法的主要框架：

```
const
```

```
    MaxLen = 75; (单词的最大长度)
```

```
type
```

```
    Twork = object
```

```
        Top, Max: Integer;
```

```
        (Top 表示堆栈中元素的个数, Max 表示最大词链的长度) ;
```

```
        Word, Ans: array[1..MaxLen] of String[MaxLen];
```

```
        (栈中每一层记录的词)
```

```
    procedure Work;
```

```
end;
```

```
procedure Twork.Work;
```

1.  $Top \leftarrow 0; Max \leftarrow 0;$
2.  $Readln(St)$ ; 读入一个词
3. While  $St \neq '.'$  Do
  - While  $(Top > 0)$  and  $(Pos(Word[Top], St) \neq 1)$  do  $Top \leftarrow Top - 1;$   
查找堆栈中的词判断哪些词语能够继续和目前的这个词构成词链
5.  $Top \leftarrow Top + 1; Word[Top] \leftarrow St;$
6. If  $Top > Max$  then
7.     begin  $Max \leftarrow Top; Ans \leftarrow Word;$  end; 更新 Max 的值
8.  $Readln(St)$ ; 读入一个词
9. 输出

附具体程序编码如下:

```

program Problem_1;
const
  MaxLen = 75;
  InputFileName = 'Input.Txt';
  OutputFileName = 'Output.Txt';

var
  Answer, Word: array [1 .. MaxLen] of String[MaxLen]; {答案和堆栈}
  TestFile: Text;
  St: String[MaxLen]; {读入的单词}
  Top, Max: Integer; {堆栈中单词的个数和最长词链的长度}
  Buf: array [1 .. 4095] of Char;

Begin      {主程序}
  Assign(TestFile, InputFileName);
  SetTextBuf(TestFile, Buf);
  Reset(TestFile);
  Readln(TestFile, St);
  while St <> '.' do
    begin
      while (Catena > 0) and (Pos(Word[Top], St) <> 1) do Dec(Top); {堆栈进行回溯}
      Inc(Top);
      Word[Top] := St; {进栈}
      if Top > Max then
        begin Max := Top; Answer := Word; end; {调整最优解}
      Readln(TestFile, St);
    end;
  Close(TestFile);
  Assign(TestFile, OutputFileName);
  Rewrite(TestFile);
  for Catena := 1 to Max do Writeln(TestFile, Answer[Catena]); {输出}
  Close(TestFile);

```

end.