

題目：

以下的一段英文是節錄自 *Thinking in Java*, 2nd edition, Revision 12, 2000 by Bruce Eckel。

But computers are not so much machines as they are mind amplification tools (“bicycles for the mind,” as Steve Jobs is fond of saying) and a different kind of expressive medium. As a result, the tools are beginning to look less like machines and more like parts of our minds, and also like other forms of expression such as writing, painting, sculpture, animation, and filmmaking.

1. 先將上述的文章存入一維字元陣列 `tmp` 中。

```
char tmp[1000] = "But computers are ..."
```

再用 Structure Definition，定義一個新的型別〔名為 `keyWord`〕。其中一個 member 為長度為 50 的字元陣列，另一個 member 型別為 `int`。如下段程式碼所示：

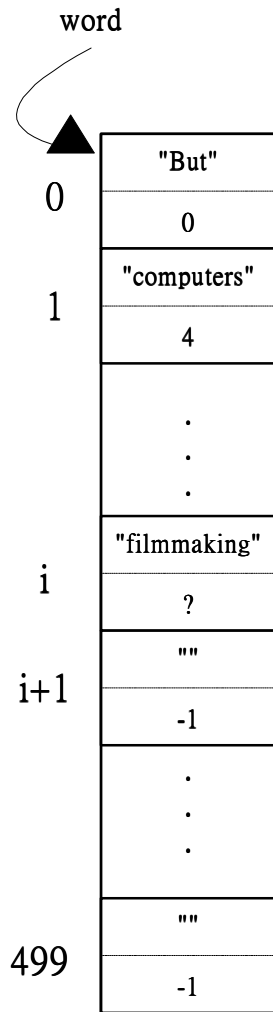
```
struct keyword{
    char str[50];
    int index;
};

typedef struct keyword keyWord;
```

在定義完 `keyWord` 型別後，再宣告一個陣列 `word` 儲存那一段文字之相關資訊。

```
keyWord word[500];
```

且將每一個 `word` 元素中的 `str` 字串初始化為空字串，`index` 初始化為 `-1`。如下所示：



其中，最後一個存入 word array 的單字〔即 filmmaking〕，在 word array 中的 index 為 "i"。

最後，將 word 陣列的內容列印在螢幕上，格式如下：

Before sorting:

	str	index
word[0]:	But	0
word[1]:	computers	4
	.	
	.	
	.	

```
word[i]:      filmmaking      *
```

其中，最後一個存入 word array 的單字〔即 filmmaking〕，在 word array 中的 index 爲” i”、開頭字元在 tmp 中的 index 爲” *”。

2. 根據上題所得之陣列 word 做 bubble sort (參考課本 220 頁)，使 keyWord 元素能有”順序”〔稍後定義〕地排列。亦即：在排列完後

$$\text{word}[0] < \text{word}[1] < \dots < \text{word}[i]$$

其中任兩個 keyWord 元素比較的方法如下：

- (1) 在得到任兩 keyWord 元素〔例 $\text{word}[j], \text{word}[k]$ 〕時，先比較兩元素中各自的 str member〔即 $\text{word}[j].\text{str}$, $\text{word}[k].\text{str}$ 〕，字串中所有英文字母的均視爲小寫。例如：

”IS”、”Is”或”iS”均視爲與”is”相同

再使用 strcmp 來決定此二字串的順序，此二字串會有下列三種可能順序。下面針對每一可能定出其結果：

- (1.1) if $\text{word}[j].\text{str} > \text{word}[k].\text{str}$, then $\text{word}[j] > \text{word}[k]$
- (1.2) if $\text{word}[j].\text{str} < \text{word}[k].\text{str}$, then $\text{word}[j] < \text{word}[k]$
- (1.3) if $\text{word}[j].\text{str} = \text{word}[k].\text{str}$, then 執行下一步驟〔步驟(2)〕

- (2) 若 $\text{word}[j].\text{str} = \text{word}[k].\text{str}$ ，就再取出各自的 index member〔即 $\text{word}[j].\text{index}$, $\text{word}[k].\text{index}$ 〕來做比較，以決定 $\text{word}[j]$ 和 $\text{word}[k]$ 的順序。以下有兩種情形：

- (2.1) if $\text{word}[j].\text{index} > \text{word}[k].\text{index}$, then $\text{word}[j] > \text{word}[k]$
- (2.2) if $\text{word}[j].\text{index} < \text{word}[k].\text{index}$, then $\text{word}[j] < \text{word}[k]$

最後，排列後的 word 陣列輸出至螢幕上。其格式如下圖：

After sorting:

	str	index
word[0]:	a	?

```
word[1]:    a                ?
word[2]:    amplification    ?
           .
           .
           .
word[i]:    writing           ?
```

3. 請寫一程式，可讓使用者輸入一個任意的英文單字，程式會用 `binarySearch` 的方式(參考課本 224 頁)，去搜尋該單字是否出現在 `word` 陣列中，並印出其所有出現的 `index`。

其最終輸出格式如下：

```
Enter string search key: tool
Value not found.
```

```
Enter string search key: Are
```

```
                str                index
-----
word[?]    are                ?
word[?]    are                ?
           .
           .
           .
word[?]    are                ?
```