Counting Words

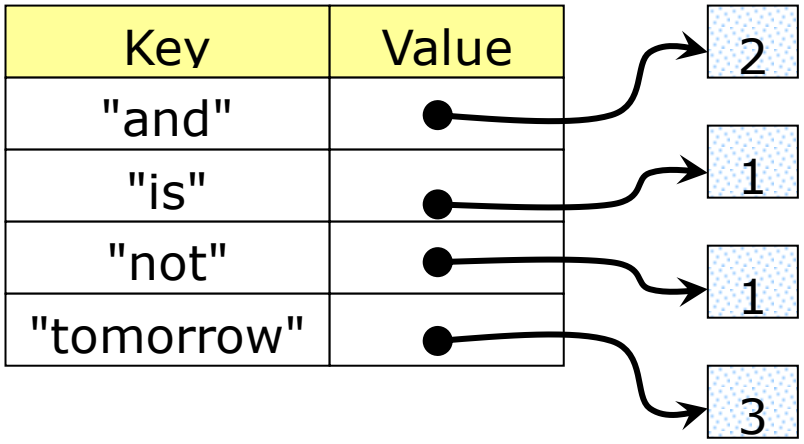
We start this lecture by looking at a 'not-very-useful' application program that counts how often a word appears in the input, ***one word per line***.

Data:

tomorrow
and
tomorrow
and
tomorrow
is
not

Output:

and	2
is	1
not	1
tomorrow	3



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "symtab.h"
typedef struct {int count;} *counterT;
main() {
   char line[80];
   symtabADT table;
   table = EmptySymbolTable();
   scanf("%s", line);
   while (strcmp(line, "***")!=0) {
      RecordWord(table, line); scanf("%s", line);
   }
   DisplayWordFrequencies(table);
}
```

Note

- **#include <stdio.h>**
  #include <stdlib.h>
  #include <string.h>
  #include "symtab.h"
  We need to use scanf, printf, and NULL, right? And we also need to use a symbol table, and strings as well.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "symtab.h"
typedef struct {int count;} *counterT;

main() {
    char line[80];
    symtabADT table;
    table = EmptySymbolTable();
    scanf("%s", line);
    while (strcmp(line, "***")!=0) {
        RecordWord(table, line);
        scanf("%s", line);
    }
    DisplayWordFrequencies(table);
}
```

- **char line[80];**
  We set a maximum length for the line.

- **symtabADT table;**
  This is the symbol table that we use to store the frequencies of characters.

- **table = EmptySymbolTable();**
  We get a brand new symbol table to use.

- **scanf("%s", line);**
  Then we read in the line (one word).

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "symtab.h"
typedef struct {int count;} *counterT;

main() {
    char line[80];
    symtabADT table;
    table = EmptySymbolTable();
    scanf("%s", line);
    while (strcmp(line, "***")!=0) {
        RecordWord(table, line);
        scanf("%s", line);
    }
    DisplayWordFrequencies(table);
}
```

- **while (strcmp(line,"***")!=0)**
  {
  RecordWord(table, line);
  scanf("%s", line);
  }

  We call RecordWord(table, i^{th} line) until end of input (indicated by string **"***"**).

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "symtab.h"
typedef struct {int count;} *counterT;

main() {
    char line[80];
    symtabADT table;
    table = EmptySymbolTable();
    scanf("%s", line);
    while (strcmp(line, "***")!=0) {
        RecordWord(table, line);
        scanf("%s", line);
    }
    DisplayWordFrequencies(table);
}
```

```
void RecordWord(symtabADT table, char *word) {

   counterT entry;

   entry = Lookup(table, word);
   if (entry==NULL) {
      entry = (counterT) malloc(sizeof(*entry));
      entry->count = 0;
      Enter(table, word, entry);
   }

   (entry->count)++;
}
```

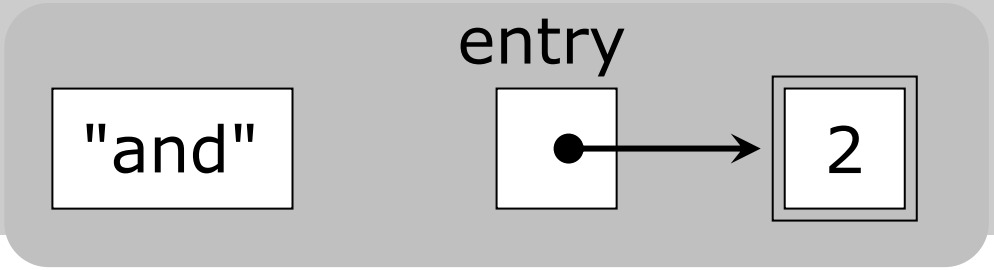
Note

- **entry = Lookup(table, word);**
  Does an entry with key=**word** exist in **table**?

```
void RecordWord(symtabADT table, char *word) {

    counterT entry;

    entry = Lookup(table, word);
    if (entry==NULL) {
        entry = (counterT) malloc(sizeof(*entry));
        entry->count = 0;
        Enter(table, word, entry);
    }

    (entry->count)++;
}
```

typedef struct {int count} *counterT;

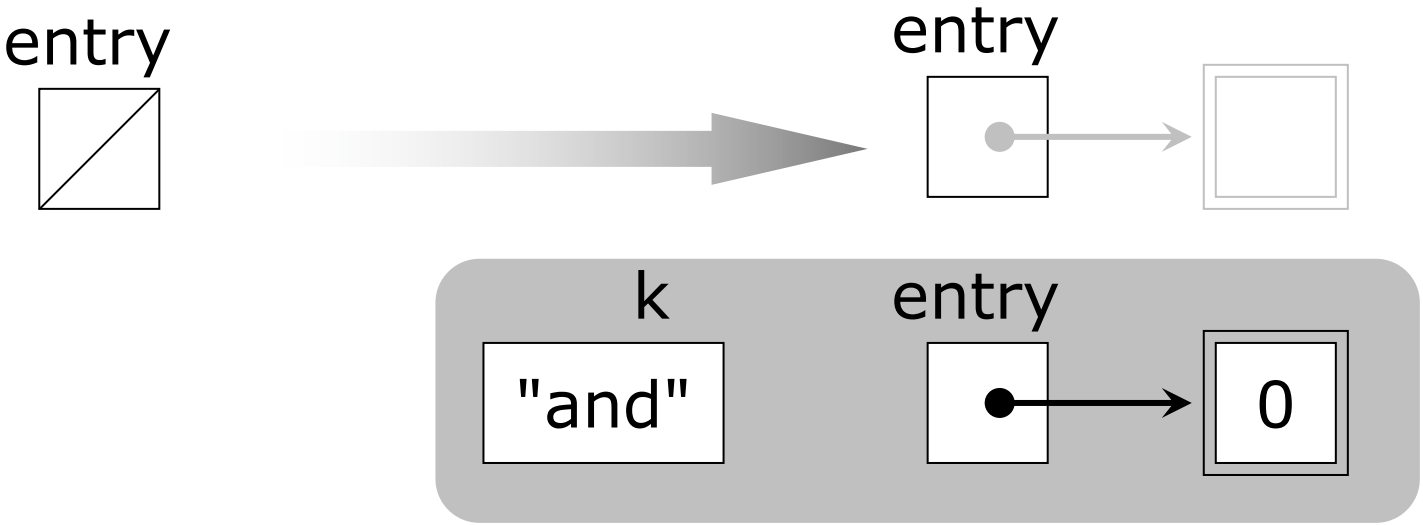
An ‘entry’ of "and":



- **if (entry==NULL) {**
  entry =
  (counterT)
  malloc(sizeof(*entry));
  entry->count = 0;
  Enter(table,word,entry);
  }

If no existing entry, create a new one and enter it to the table.

```
void RecordWord(symtabADT table, char *word) {

    counterT entry;

    entry = Lookup(table, word);
    if (entry==NULL) {
        entry = (counterT) malloc(sizeof(*entry));
        entry->count = 0;
        Enter(table, word, entry);
    }

    (entry->count)++;
}
```



- **(entry->count)++;** Finally, increment the **count**.

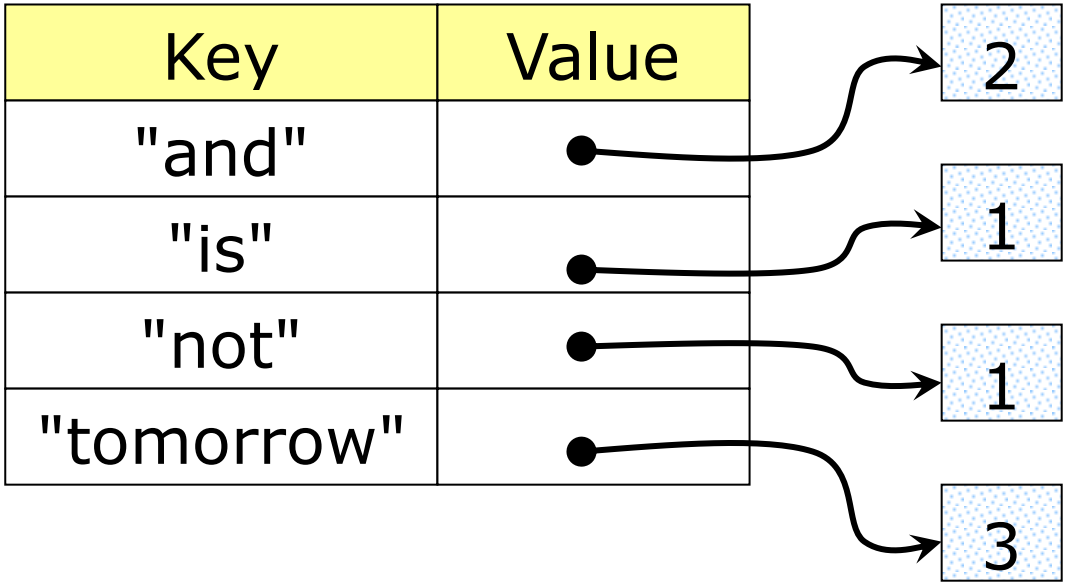
```
void RecordWord(symtabADT table, char *word) {

    counterT entry;

    entry = Lookup(table, word);
    if (entry==NULL) {
        entry = (counterT) malloc(sizeof(*entry));
        entry->count = 0;
        Enter(table, word, entry);
    }

    (entry->count)++;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "symtab.h"
typedef struct {int count;} *counterT;

main() {
   char line[80]; symtabADT table;
   table = EmptySymbolTable();
   scanf("%s", line);
   while (strlen(line)>0) {
      RecordWord(table, line); scanf("%s", line);
   }
   DisplayWordFrequencies(table);
}
```

We have a serious problem with

DisplayWordFrequencies(table);

We do not know how to write this function because we do not know what entries are there in the table!



With a table, we can **ONLY**

- Enter an entry;
- Look up an entry

But we ***cannot*** do the following

for (*every entry in the table*) {
***Display the key and the corresponding value*;**
}

Because we do not know what entries there are in the table.

We do not know how a table is implemented.

We can only use the operations provided (*e.g.*, Enter, Lookup, *etc*.) to manipulate the table.

What Should We Do?