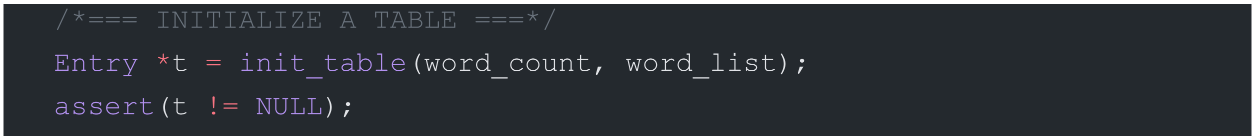
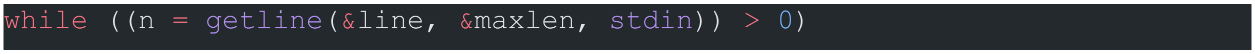
CS 433.533

Creating a dynamic table (array of structs) based on the command-line Arguments - line 30



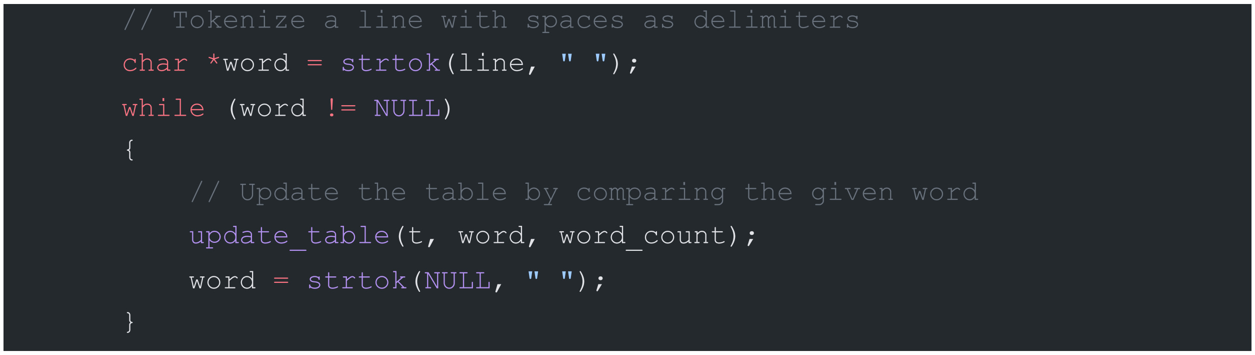
```
   /*=== INITIALIZE A TABLE ===*/
   Entry *t = init_table(word_count, word_list);
   assert(t != NULL);
```

Reading text from standard input - line 39



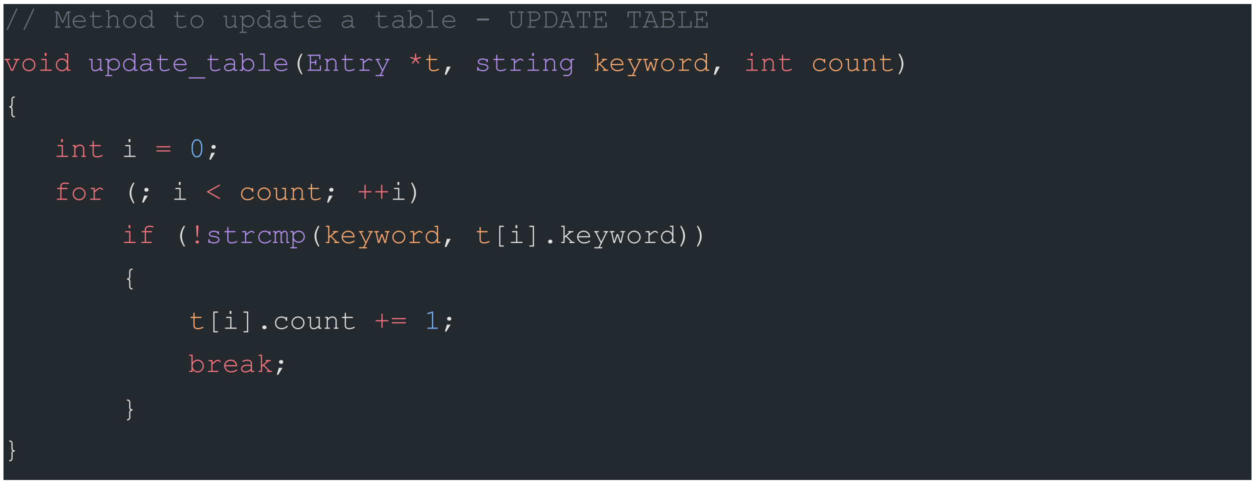
```
while ((n = getline(&line, &maxlen, stdin)) > 0)
```

Tokenizing the text and creating a dynamic array - line 44



```
       // Tokenize a line with spaces as delimiters
       char *word = strtok(line, " ");
       while (word != NULL)
       {
           // Update the table by comparing the given word
           update_table(t, word, word_count);
           word = strtok(NULL, " ");
       }
```

Comparing keywords and updating keyword count - line 79



```
// Method to update a table - UPDATE TABLE
void update_table(Entry *t, string keyword, int count)
{
   int i = 0;
   for (; i < count; ++i)
       if (!strcmp(keyword, t[i].keyword))
       {
           t[i].count += 1;
           break;
       }
}
```

CS 433.533

Display the keyword/count table - line 91

```
// Method to display a table - DISPLAY TABLE
void display_table(Entry *t, int count)
{
   if (count > 0)
       printf("Here is the number of times each keyword appears:\n");

   int i = 0;
   for (; i < count; ++i)
       printf("%s: %d\n", t[i].keyword, t[i].count);
}
```

Use of suitable functions to perform the different tasks above - line 16

```
/* Methods and operations on a table of entries */
Entry *init_table(int count, string *keywords);
void update_table(Entry *t, string keyword, int count);
void display_table(Entry *t, int count);
```

READMEfile: Provided along with code