Algorithms Worksheet 3

This assignment requires you to compare two ways of storing and retrieving data records. In order to do the comparison, you must write a program which stores data records in a linked list and a hash table with direct chaining as collision resolution strategy. The data to be used for this assignment is the story "A Portrait of the Artist as a Young Man" by James Joyce. You can download it and save it as a text file. Your program must take the file name as an argument to main() so if your program is called foo.c and is compiled to foo and the text file is called a_portrait.txt, you would call your program using

```
foo a_portrait.txt
```

When your program has read the input text file, it is expected to answer queries via a user interface. For a query, the user types in a keyword. To answer a query, your program prints the number of occurrences of the given keyword in the input text file and possibly some other statistics. Hence

```
>./foo example.txt

Reading into a list took 1.05s for 44 words.
Reading into a table took 0.54s for 44 words.

Enter word for retrieval: time

’time’ occurs 3 times.
Retrieval from list took 9 comparisons.
Retrieval from table took 3 comparisons.

Another query? [y/n] y
Enter word for retrieval: your

’your’ occurs 2 times.
Retrieval from list took 19 comparisons.
Retrieval from table took 2 comparisons.

Another query? [y/n] n
>
```

[The information about the number of comparisons is an optional extra]

Program Design

Choose a systematic approach when designing your program: Define the data structures that you need. Define the functions that you need to operate on these data structures. Re-use types and functions wherever possible. If a problem is too difficult to solve in one go, divide and conquer. A good starting point is to first develop a program that uses the linked list approach - a sample version of this has been supplied. Several functions of this code can be re-used when extending the program to use a hash table. (Note: You must hand in ONE program only; not one for the linked list and another one for the hash table!) A good solution will have at least 6 functions in addition to the main program: A function that reads the data. A function that inserts data into the linked list. A function that retrieves data from the linked list. A function that inserts data into the hash table. A function that retrieves data from the hash table. A function that handles the queries. Choose the hash function and the size of the hash table carefully. You will have to set some conventions e.g. for dealing with capitalised words, or phrases like "I'll" and "we've". There is not a single right solution. Choose something which seems sensible to you.

Assessment

You will have to mark your assignment yourself, using the marking scheme provided on the marksheet. These marks will be checked.

Submission

Your submission should contain:

The source code of your program.

Sample output showing:

The time taken to organise the data in a) the linked list and b) the hash table; and the total number of words read.

It might give the size of the hash table, the number of empty indexes, what percentage of the hash table is used (i.e. not empty) and the maximum length of an overflow list.

It should list the number of occurances of the words

looking, me, epiphany, seraphim, ardent, windless, esthetic, computer.

in the text. It might also give the number of comparisons necessary to retrieve the words.

The sample output should be submitted in a file called OUTPUT.txt.

You may submit a short report explaining anything you might think would help with the marking of an unfinished project.

A marksheet where you have filled in the blanks. Download the marksheet from the github and edit it.

Some suggest code for loading the file has been included in the folder.