

HW #2 (Word locator)

- In this assignment, you are to develop a program written in C++ (or Java if you prefer), which will allow a user to check if a specified (re)occurrence of a specified query word appears in the input text file.
- When started, your program will display a prompt ">" (printed on *stdout/cout*) and will then be ready to accept one of the following commands:

HW #2 (2)

1. “**load** <*filename*>”: This command loads the specified file.
 - A word in this file is defined as a sequence of upper and lower case letters in the English alphabet (i.e., characters ‘a’ to ‘z’, and ‘A’ to ‘Z’), numbers, and the **apostrophe**. All other characters are considered as white space and will therefore be treated as terminating a word.
 - Two successive load commands should be treated as if there is an intermediate “**new**” command (see below) in between the two commands.

HW #2 (3)

2. “**locate <word> <n>**”: This command outputs the number of the word, counting from the beginning of the file, of the *n*th occurrence of the word. Word numbering starts from 1, so the first word in the load file has a word number of 1. *The locate command is case sensitive*, i.e., to match the word in the locate command with a word in the load file you should use a case-sensitive string comparison method. If there are no matches for the locate command, print “**No matching entry**”.
- The syntax of the locate command is “locate <word> <n>”. The “<word>” parameter will have a whitespace before and after it, and “<n>” should be an integer greater than 0.

HW #2 (4)

- As an example, the following are legal commands:
“locate sing 3” and “locate sing 3”
- Both locate the 3rd occurrence of “sing”, but the second command has a few additional blank spaces around the parameter “sing”.
- The following commands are **not** legal: “locate sing3”, “locate sing 3q”. The first command does not specify a parameter <n>, and in the second command the parameter <n> is not an integer.
- Please note that the command “locate sing23 4” is a legal command for locating the fourth occurrence of the word “sing23”.

HW #2 (5)

3. “**new**”: This command resets (i.e., erases) the word list to the original (empty) state.
4. “**end**”: This command terminates the program.
 - Your program should respond to incorrect commands in the following ways:
 - If a bad command is entered, print the precise string “**ERROR: Invalid command**”, and go to the next prompt.
 - Examples of bad commands are: “**find** word 7” and “locate song”. Other examples of bad command include the locate command having a word that is **not** legal as per the definition above. For example “ra#s” and “rats!” are invalid word parameters.

HW #2 (6)

- Note that if an incorrect load command is entered, such as “load” (no filename) then your word list should **not** be reset (or erased). In other words, if you have a previously loaded file, subsequent locate commands should still query that previously loaded file.
- Similarly, if the load command specifies an invalid file name, then you should **not** reset (or erase) the word list. In both cases of the invalid load command outlined above, please print the standard error message **“ERROR: Invalid command”**.

HW #2 (7)

- If there is extraneous content in the command, such as “locate word 5 17” or “new 12”, print out the standard error message: “**ERROR: Invalid command**”.
- All the command keywords are case **sensitive**, so “LoCATE sing 2” is an invalid command.
- **Example**
- Given the following sample text file, **sixpence.txt**:

HW #2 (8)

Sing a song of sixpence,
A pocket full of rye;
Four and twenty blackbirds
Baked in a pie.
When the pie was opened,
They all began to sing.
Now, wasn't that a dainty dish
to set before the King?
The King was in the
countinghouse,

Counting out his money;
The Queen was in the
parlor
Eating bread and honey.
The maid was in the
garden,
Hanging out the clothes.
Along there came a big
black bird
And snipped off her nose!

HW #2 (9)

- The following is a sample run:

```
> load sixpence.txt
```

```
> locate song 1
```

```
3
```

```
> locate pie 1
```

```
18
```

```
> locate pie 2
```

```
21
```

```
> locate pie 3
```

```
No matching entry
```

HW #2 (10)

> locate prince

ERROR: Invalid command

> locate prince 1

No matching entry

> new

> locate song 1

No matching entry

> end