Programming Assignment #5

Word sequences from a file

CS 3358.751, Summer II 2013
Instructor: Jill Seaman

Due: Thursday, 8/1/2013 (upload electronic copy by 11:30am)

Problem:

For this assignment you will write a program that produces a list of **all the n-word sequences** that occur in a text document. A **word** is a sequence of characters containing no whitespace (**whitespace** includes spaces, tabs, and newlines). An **n-word sequence** is just a list of n words that occur in that order in the document.

For example, if the document starts this way:

```
Death of a Salesman:
In the play, Arthur Miller’s Death of a Salesman: Willy Loman, a sympathetic
salesman and despicable father who’s “life is a casting off” has some traits that
match Aristotle’s views of a tragic hero. Willy’s series of “ups and downs” is identical
to Aristotle’s views of proper tragic figure; a king with flaws. His faulty personality,
the financial struggles, and his inability are three substantial flaws that contribute to
his failure and tragic end.
```

The list of 6-word sequences would start out this way:

```
Death of a Salesman: In the
of a Salesman: In the play,
a Salesman: In the play, Arthur
Salesman: In the play, Arthur Miller’s
In the play, Arthur Miller’s Death
the play, Arthur Miller’s Death of
play, Arthur Miller’s Death of a
Arthur Miller’s Death of a Salesman:
Miller’s Death of a Salesman: Willy
Death of a Salesman: Willy Loman
...
```

Your program should take the name of a file and the value for n (the number of words in a sequence) as input. Then your program should output the list of all the n-word sequences to the screen (or a separate text file). I recommend combining the n words into a single string before outputting (this will help in the next assignment).

Your program should only read in as little of the file as necessary before it begins its output. In other words, your program should NOT read the entire file into memory before processing it. It should store not more than n+1 words in memory at once.

Other requirements:

- Newlines should not be included in your sequences (newlines from the input text should NOT be part of the output)
- Output the sequences in the same order that they were read from the file.
- Do not simply break the file up into n-word segments that, when appended will re-create the file. In other words, this output is incorrect:

```
Death of a Salesman: In the
play, Arthur Miller’s Death of a
Salesman: Willy Loman, a sympathetic salesman
. . .
```

NOTES:

- Just one *.cpp file.

- I (strongly) recommend using a queue to assist with producing the sequences from the file. You will need a queue that will allow you to enqueue, dequeue, and then access a specific element by index (operator[]). There is a data structure in the STL that allows you to push_back and pop_front, and access each element using the square bracket notation (queue[10]). It is called a deque (“deck”).

- The next assignment (PA#6) will use the solution from this assignment.

Style: See the Style Guidelines document on the course website.

Logistics:

Please submit you solution in a single file. You can call it process_files_xxxxxx.cpp. The xxxxx is your TX State NetID (your txstate.edu email id).

Submit: an electronic copy only, using the Assignments tool on the TRACS website for this class.