Paul Vorobyev (pv149) and Vishal Rohra (vlr36)
Systems Programming
Section 06
Assignment 0 - Pointer Sorter

Our program extracts alphabetic tokens (words) from a string and sorts them lexicographically. It extracts tokens by keeping the place of the current token and traversing the string until a non-alphabetic character is encountered. Once an entire word is found, `memcpy()` is called to copy it into a new `char *` variable and it is then added to the beginning of a linked list -- which is defined as `typedef struct Node {...} Node` on line 17. This process is repeated until the entire string has been traversed.

Afterwards, the linked list is converted into an array before sorting. This intermediary step may seem superfluous, but skipping the linked list and creating an array outright would also require an "extra pass" through the string to be able to `malloc()` an array of an appropriate size. For the sake of saving space, each Node in the linked list is freed after use.

Once the linked list has been converted to an array, it is sorted in place using selection sort. This algorithm will sort the array in `O(n^2)` time.