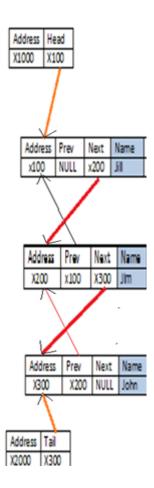Write a program that reads a list of students (first names only) from a file. It is possible for the names to be in unsorted order in the file but they have to be placed in sorted order within the linked list.

The program should use a **doubly linked list**.

Each node in the doubly linked list should have the student's name, a pointer to the next student, and a pointer to the previous student. Here is a sample visual. The head points to the beginning of the list. The tail points to the end of the list.

When inserting consider all the following conditions:

```
if(!head){                              //no other nodes
}else if (strcmp(data, head->name)<0){   //smaller than head
}else if (strcmp(data, tail->name)>0){   //larger than tail
}else{                                   //somewhere in the middle
}
```

```
When deleting a student consider all the following conditions:
     student may be at the head, the tail or in the middle
```

Below, you will find a sample of what the file looks like. Notice the names are in unsorted order but the information placed in the linked list (above visual) is in sorted order. The name of the file should be "input.txt".

In the text file, the word delete followed by a name, should delete the node with that specific student's name from the doubly linked list. If the name is not found, then nothing is deleted.

(NOTE: The above visual represents only the first three lines from the text file below.)

Jim
jill
John
delete jill
Bob
Jack
delete jim

At the end of the program, traverse through the contents of the linked list in both ascending and descending order, using the doubly linked list, and write the contents into the file output.txt. For example, given the above list, here is the sample display:

Bob
Jack
John
=============
John
Jack
Bob