Project PC25 Ch5 Write-Up

**Meena Chandok**

**Pseudo-code**

1. Included <fstream> - To open the file for reading and writing
2. Include <string> for string
3. Variable declaration and Value assignment for variables. ifstream infile; string StudentName;

string FirstStudentAlpha; string LastStudentAlpha; int Counter and initialized the counter to 0;

1. To open and read the file LineUp.txt - define the path to the location of the file.
2. The counter is initialized to ‘0’ to begin with. Since this value will remain unchanged if the file is not read and will not go through the ‘while’ loop, it serves as a flag for “file could not be read/file read incorrectly” and displays the same message at the end of the program.
3. The ‘While’ loop is used to sequentially read each name in the file. When the first name is read, it is relegated to the beginning as well as the end of the list. Every subsequent name is compared alphabetically to the beginning and end of the list. The first name on the list is replaced with the current name if it precedes it alphabetically. If the last name in the list precedes the current name, it is replaced by the current name. If neither of these situations is true, the first and last names on the list are retained.
4. If the list was read correctly and the first name was read correctly, the counter gets incremented. The names are compared as described in the previous step. The results obtained, i.e. the first and last names in the list are displayed. As stated earlier, if the file could not be read, the counter does not increment and the ‘else’ branch displays that the “file could not be read”.
5. If the file contains name(s) that begin with both upper and lower case letters, the lower case names default to the later positions as expected, based on their ASCI values.
6. If file contains names with a space in between, the names will be treated as two separate names. It is unclear what the equivalent of a ‘getline’ command is for reading from the txt File.

**Test Cases:**

**Valid file:** If the file exists (in the specified location), it will open and read. If the file name is incorrect, it will not be read.

**Valid Data:** All the names in lower case or upper case without space within each name.

**Invalid Data:** Empty file, mixed names, numbers, symbols.

**Test Case Artifacts:**

|  |  |  |
| --- | --- | --- |
| **Valid file** | **Expected Output** | **Actual Output** |
| File “LineUp.txt” | If correct file name- then open and read names. Display  Statement as:  “First student by alphabet is Cherly. “  “Last student by alphabet is  homasina.” | If correct file name- then open and read names. Display  Statement as:  “First student by alphabet is Cherly. “  “Last student by alphabet is  homasina.” |
| **Valid Data** | **Expected Output** | **Actual Output** |
| Names in the file “LineUP.txt” are all with upper case except “homasina” as lower cases alphabet. | “First student by alphabet is Cherly. “  “Last student by alphabet is  homasina.” | “First student by alphabet is Cherly. “  “Last student by alphabet is  homasina.” |
| Name “homasina” in the file “LineUP.txt” changed to “Homasina” i.e. all upper cases | “First student by alphabet is Cherly. “  “Last student by alphabet is  Zona.” | “First student by alphabet is Cherly. “  “Last student by alphabet is  Zona.” |
| To check if all the names were read in file. | To display the name of students in file, added cout <<StudentName;  Display the names of all the students in files. | To display the name of students in file, added cout <<StudentName;  Display the names of all the students in files. |
| Name “homasina” in the file “LineUP.txt” changed to “Homasina” i.e all upper cases alphabet and added additional name Aron. | “First student by alphabet is Aron. “  “Last student by alphabet is  Zona.” | “First student by alphabet is Aron. “  “Last student by alphabet is  Zona.” |
| **Invalid Data** | **Expected Output** | **Actual Output** |
| Names mixed with numbers in file. Added number 1234 after Sierra in the file. | “First student by alphabet is 1234. “  “Last student by alphabet is  homasina.” | “First student by alphabet is 1234. “  “Last student by alphabet is  homasina.” |
| Names mixed with symbol like @=? in the file. Added number @=? after Sierra in the file. | “First student by alphabet is @=?. “  “Last student by alphabet is  homasina.” | “First student by alphabet is @=?. “  “Last student by alphabet is  homasina.” |
| Empty space in between the two names in file. Added space between Sierra and Glendora. | “First student by alphabet is Cherly. “  “Last student by alphabet is  homasina.” | “First student by alphabet is Cherly. “  “Last student by alphabet is  homasina.” |
| **InValid file** | **Expected Output** | **Actual Output** |
| Changed the file name spelling  File “LinUP.txt” | Incorrect file name. Display  Statement as:  “ File could not be read” | Incorrect file name. Display  Statement as:  “ File could not be read” |
|  |  |  |

Assumptions:

I am assuming that the correct file name and path have been provided to read the file.

Learning Experience:

I learned how to open, read and extract information from a file using a ‘while’ loop.

I would really like to know how to handle a file containing names with space (e.g first name – space – last name). Is there a function similar to getline to read the names? Secondly, if the file contains upper and lower case mixed names how can I get the program to convert all the names into one format and then sort them.