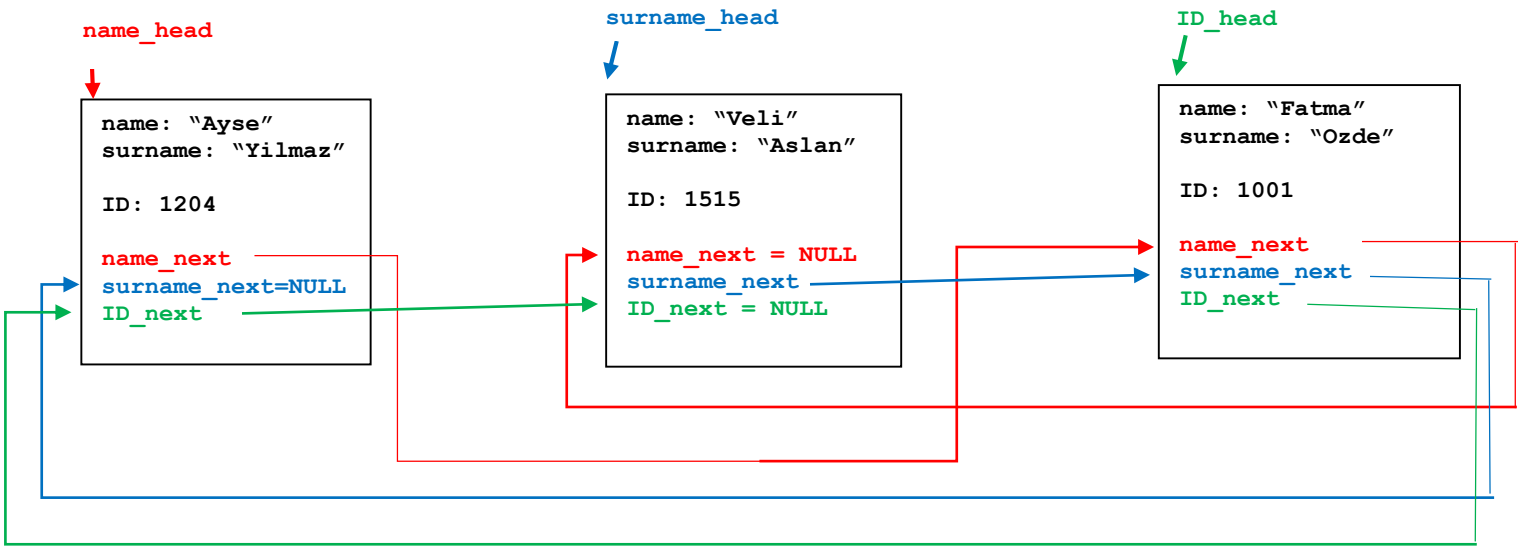
CSE 1142 - COMPUTER PROGRAMMING II
Programming Assignment #4
DUE DATE: 22/01/2021 - 23:59 (No extension)

In this assignment, you will build a linked list data structure for storing and organizing a student list that can be iterated in name-alphabetical order (sorted based on the names of the students), in surname-alphabetical order (sorted based on the surnames of the students), and in numeric order (sorted based on the student IDs). The students with their names, surnames, and IDs will be read from an input file (i.e., "**`students.txt`**") and they will be organized by using a singly linked list data structure in the program.

Each student **`struct`** in your linked list should contain a char pointer, as **`name`**; a char pointer named, as **`surname`**; a long integer, named as **`ID`**; a pointer to a **`struct`** with the same type, named as **`name_next`**; a pointer to a **`struct`** with the same type, named as **`surname_next`**; a pointer to a **`struct`** with the same type, named as **`ID_next`**. Below is a picture of the relationships between the nodes with only three students:



That's a whole lot of pointers!! The example above contains only 3 nodes (i.e., students) connected to each other with **`name_next`**, **`surname_next`**, and **`ID_next`** pointers. These pointers tie nodes to each other based on the name-alphabetical order, the surname-alphabetical order, and the ID order (increasingly). As it is seen from the example, there are 3 different types of orderings of the students. There are three head pointers (i.e., **`name_head`**, **`surname_head`**, and **`ID_head`**) that show the

starting node based on the target order. It should be noted that the head pointers can show either different or similar nodes (or students) based on the target ordering.

Firstly, you should construct the linked list (and the connections) based on the given input file. Then, we can insert a new student to the list or delete an existing student from the list.

- When we add a new node to the list, it must be inserted to the sorted list to maintain the consistent alphabetical and ID orderings.
- We can delete any existing node from the beginning, middle, or end of the list.
- After insertion and deletion operations, you should carefully re-arrange the links among the nodes.

Your tasks:

- You should declare a student **`struct`** with the mentioned features above.
- You should implement a function **`insertNode`** to insert a new node to your list. You should carefully arrange the pointers to construct different orderings.
- You should implement a function **`deleteNode`** to delete an existing node from your list. You should carefully arrange the pointers to maintain consistent orderings.
- You should implement a function **`printList`** to print the content of the linked list based three different orderings.
- You can use the linked list examples covered in lectures to implement these functions.
- Firstly, you should read the names, surnames and of the students from a given input file (i.e., **`students.txt`**).
- The input file name is given as a first command line argument to your program.
- Then, you should print the content of the list to the standard output (**`stdout`**) based on three orderings.
- Then, print a menu to the user containing options such as;
  - insert a new node to the list,
  - delete an existing node from the list,
  - print the content of the list based on the three orderings to the standard output,
  - print the content of the list based on the three orderings to a given output file, and
  - exit from the program.
- An example input file (ex: **`students.txt`**) and the produced output file (ex: **`output.txt`**) are given to you based on the sample execution scenario given below. In the input file, the name and the surname of a student is separated by a single space (consider that a student has only a single name and surname), and the ID is separated using a **`TAB`** character.

SAMPLE EXECUTION SCENARIO:

```
The list in name-alphabetical order:
      1. Ayse Yilmaz    1204
      2. Ela Kara       1980
      3. Emre Kiraz     17895
      4. Fatma Ozde     1001
      5. Ismail Celik   1345
      6. Mehmet Ari     1441
      7. Selin Ergul    24566
      8. Veli Aslan     1515

The list in surname-alphabetical order:
      1. Mehmet Ari     1441
      2. Veli Aslan     1515
      3. Ismail Celik   1345
      4. Selin Ergul    24566
      5. Ela Kara       1980
      6. Emre Kiraz     17895
      7. Fatma Ozde     1001
      8. Ayse Yilmaz    1204

The list in ID sorted order:
      1. Fatma Ozde     1001
      2. Ayse Yilmaz    1204
      3. Ismail Celik   1345
      4. Mehmet Ari     1441
      5. Veli Aslan     1515
      6. Ela Kara       1980
      7. Emre Kiraz     17895
      8. Selin Ergul    24566

Enter your choice:
   1 to insert a student into the list.
   2 to delete a student from the list.
   3 to print the students in the list.
   4 to print the students to an output file.
   5 to end.
? 1

Enter a student name, suname, and ID:
Lale Erbay  78963

Enter your choice:
   1 to insert a student into the list.
   2 to delete a student from the list.
   3 to print the students in the list.
   4 to print the students to an output file.
   5 to end.
? 3

The list in name-alphabetical order:
      1. Ayse Yilmaz    1204
      2. Ela Kara       1980
      3. Emre Kiraz     17895
      4. Fatma Ozde     1001
      5. Ismail Celik   1345
```

```
      6. Lale Erbay     78963
      7. Mehmet Ari     1441
      8. Selin Ergul    24566
      9. Veli Aslan     1515

The list in surname-alphabetical order:
      1. Mehmet Ari     1441
      2. Veli Aslan     1515
      3. Ismail Celik   1345
      4. Lale Erbay     78963
      5. Selin Ergul    24566
      6. Ela Kara       1980
      7. Emre Kiraz     17895
      8. Fatma Ozde     1001
      9. Ayse Yilmaz    1204

The list in ID sorted order:
      1. Fatma Ozde     1001
      2. Ayse Yilmaz    1204
      3. Ismail Celik   1345
      4. Mehmet Ari     1441
      5. Veli Aslan     1515
      6. Ela Kara       1980
      7. Emre Kiraz     17895
      8. Selin Ergul    24566
      9. Lale Erbay     78963

Enter your choice:
   1 to insert a student into the list.
   2 to delete a student from the list.
   3 to print the students in the list.
   4 to print the students to an output file.
   5 to end.
? 2
Enter a student ID:
24566
The student "Selin Ergul      24566" is deleted from the list!

Enter your choice:
   1 to insert a student into the list.
   2 to delete a student from the list.
   3 to print the students in the list.
   4 to print the students to an output file.
   5 to end.
? 4
Enter a file name:
output.txt
Output is printed to the file output.txt

Enter your choice:
   1 to insert a student into the list.
   2 to delete a student from the list.
   3 to print the students in the list.
   4 to print the students to an output file.
   5 to end.
? 5
--------------------------------
```

- **You have to use linked lists. Use of arrays to represent the list will not be graded.**
- **Note that there will be only a single linked list. Implementation of three lists is not allowed!**
- **It should be noted that only <u>selected parts</u> will be graded in your homework.**

Submission Instructions

Please zip and submit your files using filename YourNumberHW4.zip
(ex: 150713852HW4.zip) to Canvas system (under Assignments tab).

Your program must include necessary comments with your own words to explain your actions!

Notes:

1. Write a comment at the beginning of each program to explain the purpose of the program.
2. Write your name and student ID as a comment.
3. Include necessary comments to explain your actions.
4. Select meaningful names for your variables and class names.
5. You are allowed to use the materials that you have learned in lectures & labs.
6. Do not use things that you did not learn in the course.
7. **Program submissions** should be done through the Canvas class page, under the assignments tab. Do not send program submissions through e-mail. E-mail attachments will not be accepted as valid submissions.
8. You are responsible for making sure you are turning in the right file, and that it is not corrupted in anyway. We will not allow resubmissions if you turn in the wrong file, even if you can prove that you have not modified the file after the deadline.
9. In case of any form of **copying and cheating** on solutions, all parts will get **ZERO** grade. You should submit your own work. In case of any forms of cheating or copying, both giver and receiver are equally culpable and suffer equal penalties.
   All types of plagiarism will result in zero grade from the homework.
10. No late submission will be accepted.