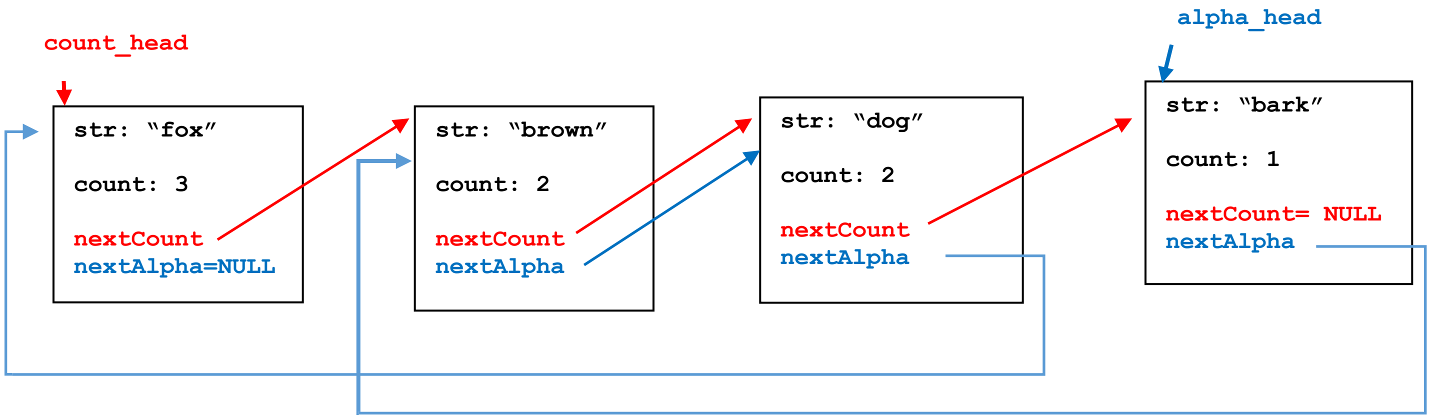
CSE 1242 - COMPUTER PROGRAMMING II
Programming Assignment # 3
DUE DATE: 31/05/2024 - 23:59 (No extension)

In this assignment, you will build a linked list data structure for storing strings. As you know, there is no String data type in C; so you will define a struct for this purpose. The input strings will be given in an input file (i.e., "**`input.txt`**") and they will be organized by using a singly linked list data structure in the program.

Each struct in your linked list should contain a char array of size **`25`**, named **`str`** that represents the string itself; an **`integer`**, named **`count`** that represents how many times that string is incremented (will be explained later); a pointer to a struct with the same type, named **`nextAlpha`** that represents the next node in alphabetical order based on the **`str`** value of the structs; and a pointer to a struct with the same type, named as **`nextCount`** that represents the next node in terms of **`count`** value of the structs in descending order. Below is an example picture of the relationships between the nodes in a list with 4 structs:



The following struct and functions must be included in your implementation.

```
typedef struct {
    …
} String;

void inc(String** headAlpha, String** headCount, char* key) {
    /* Increments the count of the string key by 1. If key does not exist in
    the data structure, insert it with count 1 and make necessary connections.
    */
}
```

```
void dec(String** headAlpha, String** headCount, char* key) {
    /* Decrements the count of the string key by 1. If the count of key is 0
    after the decrement, remove it from the data structure. It is guaranteed
    that key exists in the data structure before the decrement. */

}

char* getMaxKey(String* head) {
    /* Returns one of the keys with the maximal count. If no element exists,
    return an empty string. */
}

char* getMinKey(String* head) {
    /* Returns one of the keys with the minimum count. If no element exists,
    return an empty string. */
}
char* printList(String* head, int type) {
    /* Print the content of the linked list based on alphabetical order of the
    strings if type is 1; print the content of the list in descending order
    based on the count if the type is 2. */
}
```

Notes:

- The given key will be less than 24 characters.
- Key will contain only lower-case English letters.
- It is guaranteed that for each call to dec, key is existing in the linked list.

Suppose that the following is the content of sample input file (input.txt):

```
inc brown
inc fox
getMaxKey
getMinKey
inc dog
inc brown
inc dog
inc bark
inc barks
inc fox
inc fox
getMaxKey
getMinKey
printList 1
printList 2
dec bark
dec dog
printList 1
printList 2
```

Output:

```
brown
brown
fox
bark

The list in alphabetical order:
bark(1) --> barks(1) --> brown(2) --> dog(2) --> fox(3) --> NULL

The list in descending order:
fox(3) --> dog(2) --> brown(2) --> bark(1) --> barks(1) --> NULL

The list in alphabetical order:
barks(1) --> brown(2) --> dog(1) --> fox(3) --> NULL

The list in descending order:
fox(3) --> brown(2) --> dog(1) --> barks(1) --> NULL

----------------------------------------------------------------
```

- **You have to use linked lists. Use of arrays to represent the list will not be graded.**

- **It should be noted that only selected parts 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, you will get **FF** 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 FF grade from the course.
10. No late submission will be accepted.