

---

## CS19101 Programming and Data Structures: Third Lab Test

---

### General instruction to be followed strictly

1. Do not use any global or static variable unless you are explicitly instructed so.
2. Use proper indentation in your code and comment.
3. Name your file as <roll\_no>\_test3. For example, if your roll number is 14CS10001, then name your file as 14CS10001\_test3.c or 14CS10001\_test3.cpp as applicable.
4. Write your name, roll number, and assignment number at the beginning of your program.
5. Make your program as efficient as possible.

---

Write a program to create a linked list and sort the list. Define the structure of the node of your linked list as follows.

```
#DEFINE SIZE 50
typedef struct node{
    char first_name[SIZE];
    char last_name[SIZE];
    struct node* next;
} node;
typedef node *linkedlist;
```

### Part I: Write a function to initialize a linked list

Write a function to initialize a linked list. The prototype should be as follows which initializes the linked list pointed by l.

```
linkedlist init();
```

### Part II: Write a function to display a linked list

Write a function to display a linked list. The prototype should be as follows which displays the linked list l.

```
void display(linkedlist l);
```

### Part III: Write a function to insert in the middle of a linked list

Write a function to insert an elements in the linked list such that the newly inserted element is the middle element if the linked list has an odd number of nodes, and the left middle element if the linked list has an even number of nodes. The prototype should be as follows which inserts an integer x at the front of the linked list pointed by l.

```
linkedlist insert_middle(linkedlist l, char first[], char last[]);
```

## Part III: Write a function to delete from the middle of a linked list

Write a function to delete an element from the middle of a linked list. Delete the middle element if the linked list has an odd number of nodes, and the left middle element if the linked list has an even number of nodes.

```
linkedlist delete_middle(linkedlist l);
```

## Part IV: Sort the linked list

Write a function which takes as input a linked list and an integer, say key, as input and sorts the linked list in non-decreasing (lexicographic) order of first\_name if key is 0, in non-increasing (lexicographic) order of first\_name if key is 1, in non-decreasing (lexicographic) order of last\_name if key is 2, and in non-increasing (lexicographic) order of last\_name if key is 3.

Use the following sorting algorithm. If the last digit (from right) of your roll number, when divided by 3, gives remainder

- ▷ 0, then use insertion sort.
- ▷ 1, then use bubble sort.
- ▷ 2, then use selection sort.

The prototype should be as follows which displays the linked list l.

```
void sort(linkedlist* l, int key);
```

Submit one (single) C/C++ program.

## Sample Output

```
1. insert at middle
2. delete from middle
3. sort in non-decreasing order or first name
4. sort in non-increasing order or first name
5. sort in non-decreasing order or last name
6. sort in non-increasing order or last name
7. exit
1
Write element to be inserted: Sachin Tendulkar
Linked list: (Sachin Tendulkar,)
1
Write element to be inserted: Virat Kohli
Linked list: (Virat Kohli,Sachin Tendulkar,)
1
Write element to be inserted: Sourav Ganguly
Linked list: (Virat Kohli,Sourav Ganguly,Sachin Tendulkar,)
1
Write element to be inserted: MS Dhoni
```

Linked list: (Virat Kohli,MS Dhoni,Sourav Ganguly,Sachin Tendulkar,)

3

Linked list: (MS Dhoni,Sachin Tendulkar,Sourav Ganguly,Virat Kohli,)

4

Linked list: (Virat Kohli,Sourav Ganguly,Sachin Tendulkar,MS Dhoni,)

2

Linked list: (Virat Kohli,Sachin Tendulkar,MS Dhoni,)

2

Linked list: (Virat Kohli,MS Dhoni,)

7

---

## Policy on Plagiarism

Academic integrity is expected of all the students. Ideally, you should work on the assignment/exam consulting only the material we share with you. You are required to properly mention/cite anything else you look at. Any student submitting plagiarised code will be penalised heavily. Repeated violators of our policy will be deregistered from the course. Read this to know what is plagiarism.