CS19101 Programming and Data Structures: Linked List

General instruction to be followed strictly

1. Do not use any global variable or static variable unless you are explicitly allowed to do so.
2. Use proper indentation in your code and comment.
3. **Name your file as** `<roll_no>_<assignment_no>`**. For example, if your roll number is 14CS10001 and you are submitting assignment 3, then name your file as** `14CS10001_3.c` **or** `14CS10001_3.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.

In this program, we will implement some elementary functions of linked list. Define your linked list as follows.

```
typedef struct node{
      int data;
      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.

```
void init(linkedlist* l);
```

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 at the front of a linked list

Write a function to insert an elements at the front of a linked list. The prototype should be as follows which inserts an integer x at the front of the linked list pointed by l.

```
void insert_front(linkedlist* l, int x);
```

Part IV: Write a function to delete an element from the front of a linked list

Write a function to delete an element from the front of a linked list. The prototype should be as follows which deletes an element from the linked list pointed by l. The function should not do anything if the linked list pointed by l is empty.

```
void delete_front(linkedlist* l);
```

Part V: Write a function to reverse a linked list

Write a function to a linked list. The prototype should be as follows which reverses the linked list pointed by l. The function should not do anything if the linked list pointed by l is empty.

```
void reverse(linkedlist* l);
```

All the functions from parts III to V also displays the modified linked list. Make sure, you free any space which you do not need anymore.

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

Sample Output

```
insert_front 10: (10)
insert_front 100: (100,10)
reverse: (10,100)
delete_front: (100)
delete_front: ()
```

Policy on Plagiarism

Academic integrity is expected from 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.