

# Design of assignment 9

Struct node
Public: Data: int Node* next

Class linked_list
Public: empty(node *head): bool AddNode(node *&head, node *&tail, int num): void DelHead(node *&head, node *&tail): void Print(node *curr): void MidNode(node *curr,int length1): void length(node *&head): int

## Design:

1. bool empty(node \*head) We need to write a function which decides the linked list is empty or not then we can continue, because in different cases, when we add or delete a value from linked list, the code is different. so I create an empty(node \*head) function.
2. Void AddNode(node \*&head, node \*&tail, int num) we create a function which can add a value in the end, in this function, we need to consider 2 cases, the linked list is empty or not, if it is empty, then head and tail node just equal to temp; if it is not empty, then use tail to connect the additional node.
3. Void DelHead(node \*&head, node \*&tail) we create a function which can delete the 1<sup>st</sup> element of the linked list, we need to consider 3 cases, empty, only 1 element, not empty. (1) just print out "empty 0".(2)just delete that element. (3) create a temp node equal head, and head node point to next value, then delete temp node.

4. void Print(node \*curr) we need to consider 2 cases, empty or not, if the linked list is empty, just print out "empty 0". if not go through the linked list by a loop and print out each value each time.
5. Int length(node \*&head): this function is for determine the length of the linked list, it is similar with print function because we also need to go through the linked list. The difference is each time length+1;
6. void MidNode(node \*curr, int length1) we need to consider 2 cases in this loop, 1 value and 2 values, so we use length() function to determine the length and use "%" symbol, if remind 1 then use node go through the middle position and print out that 2 values, if remind 0 then just print out the middle value.
7. Int main() in main function, we need to transfer the string array into int array, so we use istream and getline function, then we get another string array, then we analyse the string array, if A appears then use atoi function and addnode () function, if R appears then delete the 1<sup>st</sup> number. In the end, print out the value and print out the middle value by MidNode() function.

## Test:

Input	expect
A888 A111 R A777 A5 A3	111->777->5->3 777 5
A888 A111 R A777 A5 A3 R	777->5->3 5
R A777	777 777
A888 A111 R A777 A5 A3 R R R R	Empty 0
A1 A2 A3 A4 A5 A6	1->2->3->4->5->6 3 4
A777 R	Empty 0