Assignment Solutions | Linkedlist - 1 | Week 15

1. In a singly linked list, deletion of data requires modification of how many pointers?
    1. 1
    2. 2
    3. 3
    4. Depends upon the node being deleted.

Solution : `option 2 → 2`

2. Predict the output for linked list = 1->2->3->4->5:

```
void traverse(Node* head) {
  while(head and head->next) {
    cout << head->data << ' ';
    head = head->next->next;
  }
}
```

1. 1 2 3 4 5
2. 1 3 5
3. 2 4
4. 1 3

Solution : `Option 4 → 1 3`

Q3. Implement a Linked List class.

The user defined LL should have insert (head,tail,idx) , delete(head,tail,idx) , get(idx) and display functions.

Solution :

```
#include<bits/stdc++.h>

using namespace std;

class node{

public :

int data;
```

```
node *next;
node(int n){
data = n;
next = NULL;
}
};
class linkedlist{
public:
node *head,*tail;
linkedlist(){
head = NULL;
tail = NULL;
}
void display(){
node *temp = head;
while(temp){
cout<<temp→data<<" ";
temp = temp->next;
}
cout<<endl;
}
void addFirst(int val){
node *temp = new node(val);
if(head == NULL)head = temp;
else {
temp->next = head;
head = temp;
}
if(tail == NULL)tail = head;
}
void addAtIndex(int idx , int val){
if(idx == 0)addFirst(val);
else{
```

```
idx--;
node *temp = head;
while(idx--){
temp = temp->next;
}
node *newnode = new node(val);
newnode->next = temp->next;
temp->next = newnode;
}
}
void getAtIndex(int idx){
if(idx == 0)cout<<head→data<<endl;
else{
node *temp = head;
while(idx--)temp=temp->next;
cout<<temp→data<<" ";
}
}
void deleteAtIndex(int idx){
if(idx == 0)head = head->next;
else{
node *prev = NULL, *curr = head;
while(idx--){
prev = curr;
curr = curr->next;
}
prev->next = curr->next;
curr->next = NULL;
}
}
};
int main(){
linkedlist ll;
```

```
ll.addFirst(1);
ll.addFirst(2);
ll.addFirst(3);
ll.addFirst(4);
// ll.display();
ll.addLast(1);
ll.addLast(2);
ll.addLast(3);
ll.addLast(4);
ll.addAtIndex(3,8);
ll.addAtIndex(9,10);
ll.deleteAtIndex(9);
ll.display();
// ll.getAtIndex(9);
}
```