```
#include<bits/stdc++.h>
using namespace std;

// A linked list node defined as struct with two members
struct Node {
   int data; // contains the data represented by the node
   Node* next; //pointer to the next node
};

void print_list( Node* head)
{
	Node* iter=head;
	cout << endl << "List Elements: "<< endl;
	while(iter!=NULL)
	{
		cout << iter->data << " ";
		iter=iter->next;
	}
	cout << endl;
}
int main()
{
	//creating a linked list with n nodes and input from user
	int n; // nummber of nodes in linked list
	cout << "Enter Number of nodes:";
	cin >> n;

	Node* head = NULL; // Defining the head of the linked list, currently equated to NULL

	//Taking n inputs from user and creating a linked list
	Node* tail=NULL;
```

```
for( int i=0; i<n; i++)
{
	Node* temp= new Node();
	cin >> temp->data;
	temp->next=NULL;

	if(i==0)
	{
		head=temp;
		tail=temp;
	}
	else
	{
		tail->next=temp;
		tail=temp;
	}

}

//Printing the linked list
cout << "Elements of linked list: "<< endl;
Node*temp=head;
while(temp!=NULL)
{
	cout <<temp->data << " ";
	temp=temp->next;
}
cout << endl;

//Insertion of element at begining of linked list
Node* new_node= new Node();
```

```
cout << "Enter new data:";
cin >> new_node->data;
new_node->next=head;
head=new_node;
cout << "Inserts Node at begining"<< endl;
print_list(head);

//insert element at end
Node* temp1=head;
while(temp1->next!=NULL) // finding the last node so that we can insert a new node after it
        temp1=temp1->next;

Node* new_node1= new Node();
cout << "Enter new data:";
cin >> new_node1->data;
new_node1->next=NULL;
temp1->next=new_node1;
cout << "Inserting Node at end"<< endl;
print_list(head);

//Insert element in between, lets say we are inserting a node at 4th position
int pos=4;
Node* Temp=head;
Node* prev=NULL;
for(int i=1; i<=pos-1; i++)
{
        prev=Temp;
        Temp=Temp->next;
}
Node* node= new Node();
cout << "Enter value of new node:";
```

```
cin >> node->data;
node->next=Temp;
prev->next=node;
cout << endl;
cout << "Insert node at 4th position"<< endl;
print_list(head);

//Deletion at begining
Node* deleted=head;
head= head->next;
free(deleted);
cout << "Delete Node from begining"<< endl;
print_list(head);

//Deletion at the end
Node* deleted1=head;
prev=NULL;
while(deleted1->next!=NULL)
{
        prev=deleted1;
        deleted1=deleted1->next;
}
prev->next=NULL;
free(deleted1);
cout << "Delete Node from end"<< endl;
print_list(head);

//Deletion from in between, Let's delete the 3rd element
pos=3;
Temp=head;
prev=NULL;
```

```
	for(int i=1; i<=pos-1; i++)
	{
		prev=Temp;
		Temp=Temp->next;
	}
	prev->next= Temp->next;
	free(Temp);
	cout << "Delete Node at 3rd position"<< endl;
	print_list(head);

	return 0;
}
```