```cpp
//Singly Linked List

#include<bits/stdc++.h>
using namespace std;
class node{
public:
    int data;
    node* next;
    node(int x){
        data=x;
        next=NULL;
    }
};
void push_begin(node* &head,int x){
    node* newNode=new node(x);
    if(head==NULL){
        cout<<"Linked list is empty!!!";
        return;
    }
   newNode->next=head;
   head=newNode;
}
void push_mid(node* &head,int x,int pos){
    node* newNode=new node(x);
    if(head==NULL){
        head=newNode;
        return;
    }
    node* temp=head;
    pos=pos-2;
    while(pos--){
        temp=temp->next;
    }
    newNode->next=temp->next;
    temp->next=newNode;
}
void push_end(node* &head,int x){
    node* newNode=new node(x);
    if(head==NULL){
        head=newNode;
        return;
    }
    node* temp=head;
    while(temp->next!=NULL){
        temp=temp->next;
    }
    temp->next=newNode;
}
void delation(node* &head,int pos){
    node* temp=head;
    if(pos==1){
        head=temp->next;
    }
    else{
        pos=pos-2;
    while(pos--){
        temp=temp->next;
    }
    node* temp2=temp->next;
    temp->next=temp2->next;
    }
}
void traverse(node* head){
    while(head!=NULL){
        cout<<head->data<<"  ";
        head=head->next;
```

```
    }
    cout<<endl;
}
int main(){
node* head=NULL;
    while(1){
        cout<<"\n***MENU***\n1.Insert end\n2.Insert begin\n3.Insert mid\n4.travers \n5.delete\n6.Exist\n";
        int x;cin>>x;
        if(x==2){
            cout<<"Input elemet to insert: ";
            int a;cin>>a;
            push_begin(head,a);
        }
        else if(x==1){
            cout<<"Input elemet to insert: ";
            int b;cin>>b;
            push_end(head,b);
        }
        else if(x==3){
            cout<<"Input elemet to insert: ";
            int c;cin>>c;
            int pos;
            cout<<"Position:";
            cin>>pos;
            push_mid(head,c,pos);
        }
        else if(x==4){
            system("cls");
            cout<<"Atfer traversing:";
          traverse(head);
        }
        else if(x==5){
            int d;
            cout<<"Enetre deleting position:";
            cin>>d;
            delation(head,d);
        }
        else if(x==6){
            cout<<"Program exist...";
            break;
        }
        else{
             cout<<"Please enter a valid integer\n ";
        }
}
}


//Doubly LinkedList
#include <bits/stdc++.h>
using namespace std;
class Node {
public:
  Node* prev;
  int data;
  Node* next;

  Node(int x){
      next=NULL;
      data=x;
      prev=NULL;
  }
};
void Insert_at_front(Node* &head, int data){

  Node* newNode = new Node(data);
```

```
  newNode->next = head;
  newNode->prev = NULL;
  if (head != NULL)
    head->prev = newNode;
    head = newNode;
}
void Insert_at_after(Node* prev_node, int data){
  if (prev_node == NULL) {
    cout << "previous node cannot be null";
    return;
  }
  Node* newNode = new Node(data);
  newNode->next = prev_node->next;
  prev_node->next = newNode;
  newNode->prev = prev_node;
  if (newNode->next != NULL)
    newNode->next->prev = newNode;
}
void Insert_end(Node* &head, int data){
 Node* newNode = new Node(data);
  newNode->next = NULL;
   Node* temp = head;
  if (head == NULL) {
    newNode->prev = NULL;
    head = newNode;
    return;
  }
  while (temp->next != NULL)
    temp = temp->next;
  temp->next = newNode;
  newNode->prev = temp;
}
void Delete_node(Node* &head, Node* &del_node){
  if (head == NULL || del_node == NULL)
    return;
  if (head == del_node)
    head = del_node->next;
  if (del_node->next != NULL)
    del_node->next->prev = del_node->prev;
  if (del_node->prev != NULL)
    del_node->prev->next = del_node->next;
  free(del_node);
}

void Display_list(Node* &node){
  Node* last;
  while (node != NULL) {
    cout << node->data << "->";
    last = node;
    node = node->next;
  }
  if (node == NULL)
    cout << "NULL\n";
}

int main() {
  Node* head = NULL;
  while(1){
    cout<<"\n***MENU***\n1. Insert_at_front()\n2.Insert end()\n3.Insert_at_after()\n4.Display
list\n5.Delete_node\n";
    cout<<"Choose an option:";
        int x;
        cin>>x;
        if(x==1){
            cout<<"Input an element to Insert_at_front: ";
            int a;
```

```
            cin>>a;
            Insert_at_front(head,a);
        }
        else if(x==2){
            cout<<"Input an element to insert_end: ";
            int a;
            cin>>a;
            Insert_end(head,a);
        }
        else if(x==3){
            cout<<"Input  an element to Insert_at_after: ";
            int a;
            cin>>a;
             Insert_at_after(head,a);
        }
         else if(x==4){
            system("cls");
            Display_list(head);
        }
        else if(x==5){
             Delete_node(head, head->next->next->next->next->next);
        }
        else{
            cout<<"Please enter a valid integer\n ";
        }

  }
}


//Circular Linked list

#include<bits/stdc++.h>
using namespace std;
class node{
    public:
 int data;
 node *next;
};
 node* head;
void insert_begin(){
 node *ptr,*temp;
 int item;
 ptr = new node;
 if(ptr == NULL){
 cout<<"\nOVERFLOW";
 }
 else{
 cout<<"\nEnter the node data?";
cin>>item;
 ptr -> data = item;
 if(head == NULL) {
 head = ptr;
 ptr -> next = head;
 }
 else{
 temp = head;
 while(temp->next != head)
 temp = temp->next;
 ptr->next = head;
 temp -> next = ptr;
 head = ptr;
 }
cout<<"\nnode inserted\n";
 }
}
```

```
void insert_last(){
 node *ptr,*temp;
 int item;
 ptr = new node;
 if(ptr == NULL){
 cout<<"\nOVERFLOW";
 }
 else{
 cout<<"\nEnter Data:";
 cin>>item;
 ptr->data = item;
 if(head == NULL){
 head = ptr;
 ptr -> next = head;
 }
 else{
 temp = head;
 while(temp -> next != head)
 {
 temp = temp -> next;
 }
 temp -> next = ptr;
 ptr -> next = head;
 }
 cout<<"\n node Inserted";
 }
}
void delete_begin(){
 node *ptr;
 if(head == NULL){
 cout<<"\nUNDERFLOW";
 }
 else if(head->next == head){
 head = NULL;
 free(head);
cout<<"\n node deleted\n";
 }
 else{
 ptr = head;
 while(ptr -> next != head)
 ptr = ptr -> next;
 ptr->next = head->next;
 free(head);
 head = ptr->next;
cout<<"\n node deleted\n";
 }
}
void delete_last(){
 node *ptr, *preptr;
 if(head==NULL){
cout<<"\nUNDERFLOW";
 }
 else if (head ->next == head){
 head = NULL;
 free(head);
cout<<"\n node deleted\n";
 }
 else {
 ptr = head;
 while(ptr ->next != head){
 preptr=ptr;
 ptr = ptr->next;
 }
 preptr->next = ptr -> next;
 free(ptr);
cout<<"\n node deleted\n";
```

```
}
}
void search(){
 struct node *ptr;
 int item,i=0,flag=1;
 ptr = head;
 if(ptr == NULL){
 cout<<"\nEmpty List\n";
 }
 else{
 cout<<"\nEnter item which you want to search?\n";
 cin>>item;
 if(head ->data == item){
 cout<<"item found at location "<<i+1;
 flag=0;
 }
 else{
 while (ptr->next != head)
 {
 if(ptr->data == item){
 printf("item found at location %d ",i+1);
 flag=0;
 break;
 }
 else{
 flag=1;
 }
 i++;
 ptr = ptr -> next;
 }
 }
 if(flag != 0){
 printf("Item not found\n");
 }
 }
}
void display(){
  node *ptr;
 ptr=head;
 if(head == NULL){
 cout<<"\nnothing to print";
 }
 else{
cout<<"\n printing values ... \n";
while(ptr -> next != head){ cout<<ptr -> data<<" ";
 ptr = ptr -> next;
 }
 cout<< ptr -> data<<" ";
 }
}
void sorta(){ node *current = head, *index = NULL;
 int temp;
 if(head == NULL){
 cout<<"List is empty";
 }
 else{
 do{
 index = current->next;
 while(index != head)
 {
 if(current->data > index->data){
 temp =current->data;
 current->data= index->data;
 index->data = temp;
 }
 index= index->next;
```

```
 }
 current =current->next;
 }
 while(current->next != head);
 }
}
int main (){
 int choice =0;
 while(choice != 8){
 cout<<"\nMain Menu\n";
 cout<<"\n1.Insert begin\n2.Insert last\n3.Delete from Beginning\n 4.Delete from
last\n5.Search\n6.Show\n7.sort\n8. exit";
 cout<<"\nEnter your choice?\n";
 cin>>choice;
 if(choice==1) insert_begin();
 else if(choice==2) insert_last();
 else if(choice==3) delete_begin();
 else if(choice==4) delete_last();
 else if(choice==5) search();
 else if(choice==6) display();
 else if(choice==7) sorta();
 else if(choice==8) exit(0);
 else cout<<"Please enter valid choice";
 system("CLS");
 display();
 }
}
```