

```

#include <iostream>
using namespace std;
struct node
{
    double data;
    node * next;
};
class list
{
private:
    node * head;
    node * tail;
public:
    list ()
    {
        head = NULL;
        tail = NULL;
    }
    ~list()
    {
    }
    void insert_in_last (double
val);
    void insert_at_position(double
val,int pos);
    void insert_in_start (double
val);
    void remove_first();
    void remove_at_position(int
pos);
    void remove_last();
    void print ();
    void print_pos (int pos);
};
void list::insert_in_last(double val)
{
    node * temp = new node;
    temp->data=val;
    temp->next=NULL;
    if(head==NULL)
    {
        head = temp;
        tail = temp;
    }
    else
    {
        tail->next=temp;
        tail=temp;
    }
}
void list::insert_at_position(double val, int
pos)
{
    node*temp1=new node;
    temp1->data=val;
    temp1->next=NULL;
    if (pos==1)
    {
        if ((head == NULL)&&
(tail==NULL))
        {
            head=temp1;
            tail=temp1;
            return;
        }
        temp1->next=head;
        head=temp1;
        return;
    }
    node*temp2;
    temp2=head;
    for(int i=0; i<pos-2; i++)
        temp2=temp2->next;
    if (temp2==tail)
    {
        temp2->next=temp1;
        tail=temp1;
        return;
    }
    temp1->next=temp2->next;
    temp2->next=temp1;
}
void list::insert_in_start(double val)
{
    node * temp = new node;
    temp->data=val;
    if(tail==NULL)
    {
        head=temp;
        tail=temp;
        return;
    }
    temp->next=head;
    head=temp;
}
void list::print()
{
    node*temp;
    temp=head;
    cout<<"Data in the list : ";
    while(temp!=NULL)
    {
        cout<<temp->data<<"\t";
        temp=temp->next;
    }
    cout<<endl;
}
void list::print_pos(int pos)
{
    node*temp;
    temp=head;
    for (int i=0; i<pos-1; i++)
    {
        temp=temp->next;
    }
    cout<<"Data at position "<<pos<<" : "
<<temp->data<<endl;
}
void list::remove_first()
{
    node*temp;
    temp=head;
    head=head->next;
    delete temp;
}
void list::remove_at_position(int pos)
{
    node*temp1;
    temp1=head;
    if (pos==1)
    {
        head=head->next;
        delete temp1;
        return;
    }
    node*temp2;
    temp2=head;
    for(int i=0; i<pos-2;i++)
    {
        temp2=temp2->next;
    }
    if(temp2->next==tail)
    {
        temp1=tail;
        delete temp1;
        tail=temp2;
        temp2->next=NULL;
        return;
    }
    temp1=temp2->next;
    temp2->next=temp1->next;
    delete temp1;
}
void list::remove_last()
{
    node*temp;
    temp=head;
    while(temp->next!=tail)
    {
        temp=temp->next;
    }
    delete temp->next;
    tail=temp;
    tail->next=NULL;
}
int main()
{
    list l;
    l.insert_at_position(5.4,1);
    l.insert_in_last(25);
    l.insert_in_last(50);
    l.insert_in_last(90);
    l.insert_in_last(40);
    l.insert_in_last(55);
    l.print();
    l.insert_in_last(15);
    l.print();
    l.insert_in_start(10);
    l.print();
    l.remove_first();
    l.print();
    l.remove_last();
    l.print();
    l.remove_at_position(4);
    l.print();
    l.insert_at_position(60,3);
    l.print();
    return 0;
}

```