```
struct node *display (struct node * start)
{
        struct node *ptr;
        ptr = start;
        while (ptr! = NULL)

                print ( ptr ->data);
                ptr = ptr = next;
        }
        return start;
}


struct no       node *ptr;


struct    node  * insert -beg (struct node *start)
{
        struct   node   *new -node
        int num;
        ipo input num.
                and eded se
        new - node --> data = num.
        new - node --> next = start;
        start = new-node;
        return start;
}


struct node   * insert_end (struct node * stot)
{
        struct nod   struct node    * ptr, * new -node;
        int num;
        input and declare new -node.

        new -node -> data = num.
        new - node -> next = Noll;
        ptr = start;
```

```
while(ptr -> next !=#NULL)

    ptr = ptr ->next
    ptr ->next = new-node;
        return start;
    }

void add_after ()
{

    data=num;
    plr = start;
    preptr = ptr;

    while (preptr -> data!=val)
    {

        preplr=ptr;
        ptr = ptr-> next;
    }

    prepk -> next - new node;
    new-node ->next= ptr;
        return start;
    }

    struct node * delete -beg(*start)
    {
        struct node * ptr;
        ptr = start;

        start = start ->next;
        free(ptr);
    return start;
    }
```

```c
struct node * delete_end (struct node *start)
{
        struct node *ptr, *preptr;
        ptr = start;

        while (ptr -> next != NULL)
        {
                preptr = ptr;
                ptr = ptr -> next;
        }
        preptr -> next = NULL;
        free(ptr);
        return start;
}


struct node * delete_after (*start){

        ptr = start;
        preptr = ptr;

        while (preptr -> data != val)
        {
                preptr = ptr
                ptr = ptr -> next
        }

        preptr -> next = ptr -> next.

        free(ptr);
        return start;
}
```