

Class list

```

} private:
    node *head;
    int size;
public:
    list()
{
    head = NULL;
    size = 0;
}
void insert(int val, int pos);
}

```

↑ → onwards

Insertion at head : insert(2, 1)

size = 5

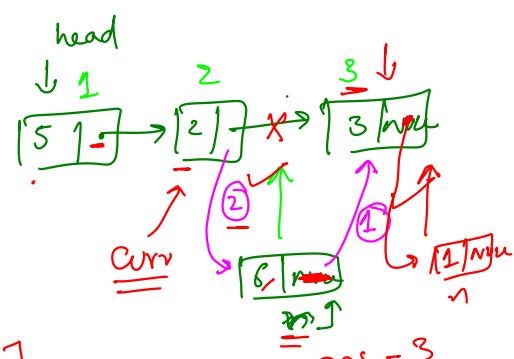
pos = 7

Insertion anywhere else :

```

void insert(int val, int pos)
{
    if (pos < 1 || pos > size + 1)
        cout << "invalid pos";
    return;
    node *n = new node(val);
    if (pos == 1) // Case 1:
        n->next = head;
        head = n;
    else
        node *curr = head;
        for (int i = 1; i < (pos - 1); i++)
            curr = curr->next;
        curr->next = n;
        n->next = curr->next;
        curr->next = n;
        size++;
}

```



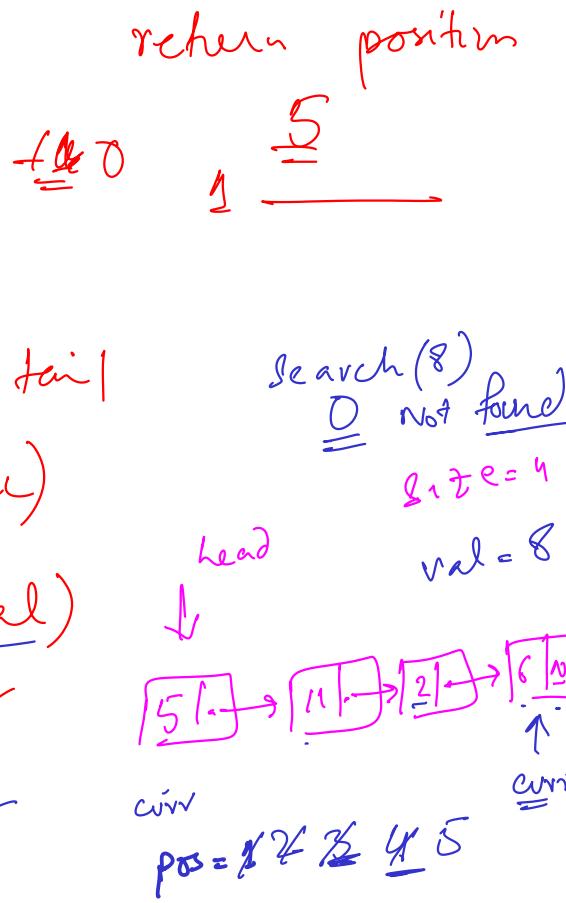
insert(1, 5)

insert(6, 3)

```

int search (int val)
{
    if (head == null)
        return 0;
    int pos = 1; ✓
    node *curr = head; ✓ tail
    while (curr != null)
    {
        if (curr->data == val)
            return pos; ✓
        curr = curr->next; ←
        pos++; ←
    }
    return 0;
}

```



```

void delete (int pos)
{
    if (pos < 1 || pos > size) ✓
        return;
    node * curr = head;
    if (pos == 1) // head
    {
        head = curr->next;
        delete curr;
    }
    else
    {
        for (int i=1; i< (pos-1); i++) curr
            curr = curr->next;
        node * delptr = curr->next;
        curr->next = delptr->next;
        delete delptr;
    }
    size--;
}

```

