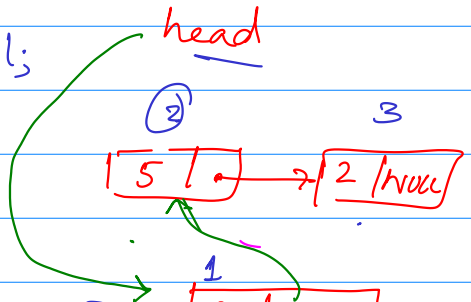
```
class LinkedList
{
  private:
      node* head;
       int  size;
  public:
      LinkedList()
      { head = NULL;
        size = 0
      }
  void insert (int val, int pos);
};
```

1 — size + 1

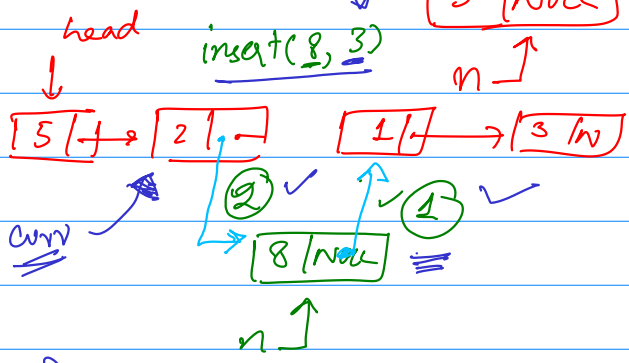
5

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

insert(3, 1)



insert(8, 3)



size = 5

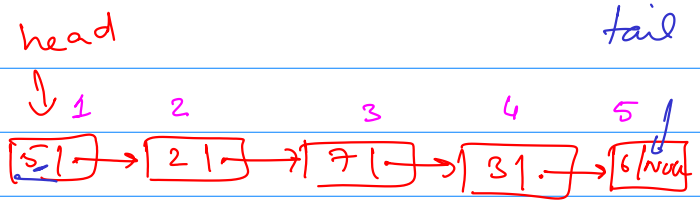
head

insert (2, 1)

Not found return 0

1 → onwards.

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



curr = NULL

val = 20, pos = 6

search (20)

0 Not found!

printList()

1 — 5 size = 6

```
void remove(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->next;
    node *delptr = curr->next;
    curr->next = delptr->next
    delete delptr;
  }
  size--;
}
```

// valid positions.